



UNIVERSIDAD DE LA RIOJA

TRABAJO FIN DE ESTUDIOS

Título

Módulo GraphicPress

Autor/es

SERGIO UGARTE SALVADOR

Director/es

LAUREANO LAMBAN PARDO

Facultad

Facultad de Ciencia y Tecnología

Titulación

Grado en Ingeniería Informática

Departamento

MATEMÁTICAS Y COMPUTACIÓN

Curso académico

2016-17



Módulo GráficoPress, de SERGIO UGARTE SALVADOR
(publicada por la Universidad de La Rioja) se difunde bajo una Licencia Creative
Commons Reconocimiento-NoComercial-SinObraDerivada 3.0 Unported.
Permisos que vayan más allá de lo cubierto por esta licencia pueden solicitarse a los
titulares del copyright.

Facultad de Ciencia y Tecnología

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática

Módulo GráficoPress

Alumno:

Sergio Ugarte Salvador

Tutores:

Laureano Lambán Pardo

Logroño, junio, 2017

Resumen

SOION, SL es una empresa que desarrolla SaaS (Software as a Service) para el sector de la construcción y gremios relacionados. El paquete software que tienen desarrollado se denomina Construpress y está compuesto por distintos módulos según las necesidades de cada empresa. El objetivo de este trabajo es desarrollar un módulo que permita comparar gráficamente los presupuestos reales, de licitación y de adjudicación de una obra, utilizando para ello la información obtenida del resto de módulos del software.

Abstract

SOION, SL is a company what develops SaaS (Software as a Service) for construction sector and related guilds. The software package which they have developed is called Construpress and it's composed by different modules according to the necessities of each company. The goal of this project is developing a module which allows us to compare graphically real budgets, bidding budgets and award budget of a building site, using the information obtained from the other modules.

Agradecimientos

Quisiera agradecer a las empresas SOION, SL y Dewenir esta oportunidad que me han dado, especialmente a mi tutor Sergio Torroba, que siempre me ha ofrecido su ayuda cuando la he necesitado. También quisiera agradecer a Daniel Tuesta y Saúl Saranova su acogida en la empresa Dewenir y la ayuda que me han proporcionado.

Agradezco también a la Universidad de La Rioja todos estos años de estudio y de preparación que me han dado, además de a mi tutor Laureano Lambán que me ha proporcionado mucha ayuda en la realización del trabajo de fin de grado.

Por último, agradecer también a mi familia por todo el apoyo que me han dado durante todos estos años de estudio.

Índice:

Capítulo 1. Introducción	4
1.1. Antecedentes	4
1.1.1. Términos	4
1.2. Objetivos	5
1.3. Tecnologías utilizadas	6
Capítulo 2. Planificación	8
2.1. Diagrama EDT del proyecto	8
2.2. Diccionario del EDT	8
2.3. Tiempos estimados	10
2.4. Diagrama de Gantt	12
2.5. Diagrama de Hitos	12
Capítulo 3. Análisis	13
3.1. Tareas previas	13
3.2. Captura de requisitos	15
3.2.1. Requisitos funcionales	15
3.2.2. Requisitos no funcionales	16
3.3. Diagrama de Casos de Uso	17
Capítulo 4. Diseño	18
4.1. Diagrama de clases	18
4.2. Diseño de la interfaz	19
4.3. Aspecto final de la aplicación	23
Capítulo 5. Implementación	29
5.1. Implementación usando el administrador	29
5.2. Implementación utilizando ficheros con código	36
Capítulo 6. Pruebas	42
6.1. Identificación de las pruebas a realizar	42
6.2. Resultado de las pruebas realizadas	43
Capítulo 7. Seguimiento y control	44
7.1. Tabla desviaciones	44
7.2. Desviaciones significativas	45
Capítulo 8. Conclusiones	47
8.1. Conclusiones	47
8.2. Posibles mejoras	47
Capítulo 9. Bibliografía	49

Capítulo 1. Introducción

1.1. Antecedentes

La empresa SOION, SL desarrolla SaaS (Software as a Service) para el sector de la construcción y gremios relacionados. El paquete software que tienen desarrollado se denomina Construpress y está compuesto por distintos módulos según las necesidades de cada empresa:

- **GuestPress:** Software para pymes y autónomos que deseen realizar de forma ágil presupuestos.
- **ControlPress:** Software para pymes y autónomos que quieran llevar un control estricto de obra, trabajadores, costes, etc.
- **TecniPress:** Software para técnicos (arquitectos, ingenieros, etc) que deseen llevar a cabo una gestión documental de las obras que están dirigiendo.
- **LicitaPress:** Software para pymes y autónomos que se dediquen a realizar licitaciones públicas o privadas, y que facilita la creación de la oferta económica final a presentar.
- **ComunicaPress:** Software para pymes y autónomos que deseen consultar mensajes y avisos relacionados con la propia empresa.
- **ManteniPress:** Software para pymes y autónomos cuyo objetivo es modificar el diseño de su cuenta dentro de la aplicación.

El proyecto que se va a realizar pretende cubrir la necesidad de desarrollo de un sistema que controle la gestión de los Presupuestos de Licitación, los Presupuestos de Adjudicación y los Presupuestos Reales de un proyecto de construcción. Por tanto, afectará a diferentes módulos de los citados anteriormente, fundamentalmente a ControlPress y LicitaPress. El objetivo es proporcionar al usuario herramientas gráficas para poder hacer un seguimiento de presupuestos que le ayuden en la toma de decisiones.

1.1.1. Términos

A continuación se explican uno a uno los términos que aparecerán a lo largo del documento:

- **Presupuesto de Licitación:** Se trata del importe que sirve de base para realizar ofertas económicas de las empresas que concurren a una licitación de obra. En particular, responde a unos costes bien tabulados y fijados con anterioridad.
- **Presupuesto de Adjudicación:** Se trata del importe que estiman las empresas para ser ofertado según el presupuesto de licitación.

- **Presupuesto Real:** Es el importe que se está acumulando durante el transcurso de un proyecto.
- **Partes de trabajo avanzado:** Son los partes de trabajo que sólo pueden ser creados por los altos cargos de una obra.

1.2. Objetivos

El producto que se pretende desarrollar con este proyecto es un nuevo módulo que se denominará BiPress. El módulo debe proporcionar información relativa a la parte económica de la empresa, en concreto a la relacionada con las obras y su ejecución. Dicha información se obtendrá a partir de los diferentes módulos del software y, fundamentalmente, tratará de proporcionar gráficas que relacionen:

- Presupuesto de Licitación vs Presupuesto de Adjudicación.
- Presupuesto de Adjudicación vs Presupuesto Real.
- Presupuesto de Licitación vs Presupuesto Real.

Además, estas gráficas se podrán descomponer por capítulos o por partidas en caso de querer conocer cuál es el desarrollo del capítulo o la partida en concreto. Le permitirá al usuario observar cómo va la facturación de la empresa en función a un presupuesto objetivo para el año en curso, de tal manera que se puedan conocer las pérdidas o ganancias de una obra en concreto, en conjunto, o en el total de la empresa.

Una vez terminado este módulo, la estructura de la aplicación tendrá el siguiente aspecto:

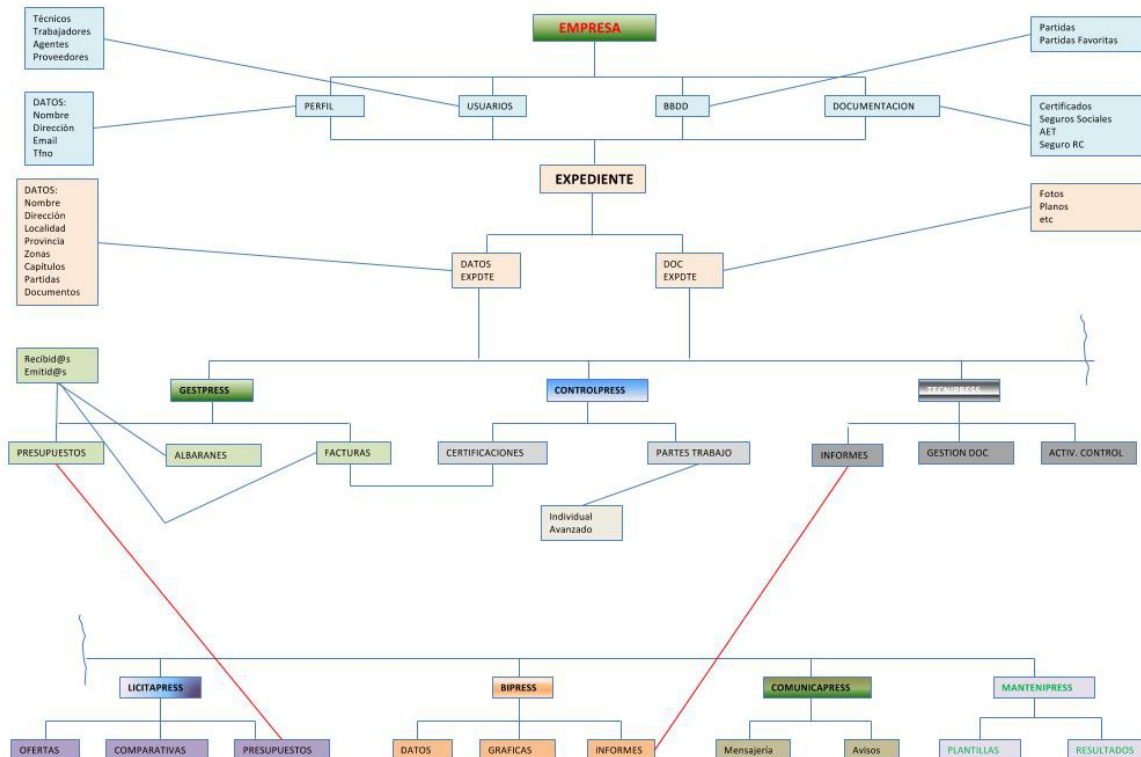


Figura 1.2.1. Estructura de la aplicación

1.3. Tecnologías utilizadas

Las tecnologías utilizadas en el proyecto han sido elegidas por la propia empresa ya que el software a desarrollar forma parte de un proyecto ya en marcha y no parece razonable la posibilidad de incorporar o modificar las tecnologías:

- **Drupal:** En su momento se eligió este sistema de gestión de contenidos (CMS) ya que se trata de un programa libre y muy configurable que permite la publicación de artículos, imágenes, archivos, pero que también ofrece la posibilidad de otros servicios añadidos. Cabe destacar también que se trata de un sistema dinámico, esto quiere decir que los contenidos son almacenados en una base de datos y se editan utilizando un entorno Web.
- **MySQL:** Se eligió este sistema gestor de bases de datos debido a su gran popularidad como base de datos, sobre todo para entornos de desarrollo web.
- **PHP (Hypertext Preprocessor):** Se ha elegido este lenguaje de programación ya que se trata de un lenguaje libre, está orientado al desarrollo de aplicaciones web dinámicas y a su compatibilidad con Drupal. Además, se considera un lenguaje bastante fácil de aprender ya que en su desarrollo se simplificaron distintas especificaciones, como es el caso de la definición de las variables primitivas, ejemplo que se hace evidente en el uso de [php arrays](#).

- **Javascript (Jquery):** Se ha elegido Jquery como librería de javascript ya que es software libre y de código abierto, y además permite la implementación tanto de proyectos abiertos como privados debido a su doble licenciamiento (Licencia MIT y la Licencia Pública General de GNU v2). Un gran beneficio que aporta Jquery es la gran variedad de funcionalidades basadas en javascript que ofrece y que de otra manera requerirían de mucho más código.
- **AJAX (Asynchronous JavaScript And XML):** Se trata de una técnica de desarrollo web utilizada para crear aplicaciones interactivas de tal manera que se permita realizar cambios en las páginas sin tener que recargarlas, mejorando la interactividad, la velocidad y la usabilidad de las aplicaciones. Además se trata de una combinación de cuatro tecnologías ya existentes: XHTML (o HTML) y hojas de estilo CSS, Document Object Model (DOM), el objeto XMLHttpRequest y XML.
- **HTML (HyperText Markup Language):** Utilizado para crear la estructura de las páginas web.
- **CSS (Cascading Stylesheets):** Utilizado para darle estilos a la estructura de HTML creado de manera que la información esté correctamente estructurada y, en combinación con las llamadas media queries, poder hacer una web responsive.

Capítulo 2. Planificación

En este apartado se mostrarán las diferentes tareas que será necesario realizar para la correcta ejecución del proyecto.

He decidido utilizar una metodología de desarrollo en cascada ya que creo que es la más adecuada para este tipo de proyecto..

2.1. Diagrama EDT del proyecto

A continuación se muestra el diagrama EDT con las diferentes tareas que componen el proyecto de una manera clara:

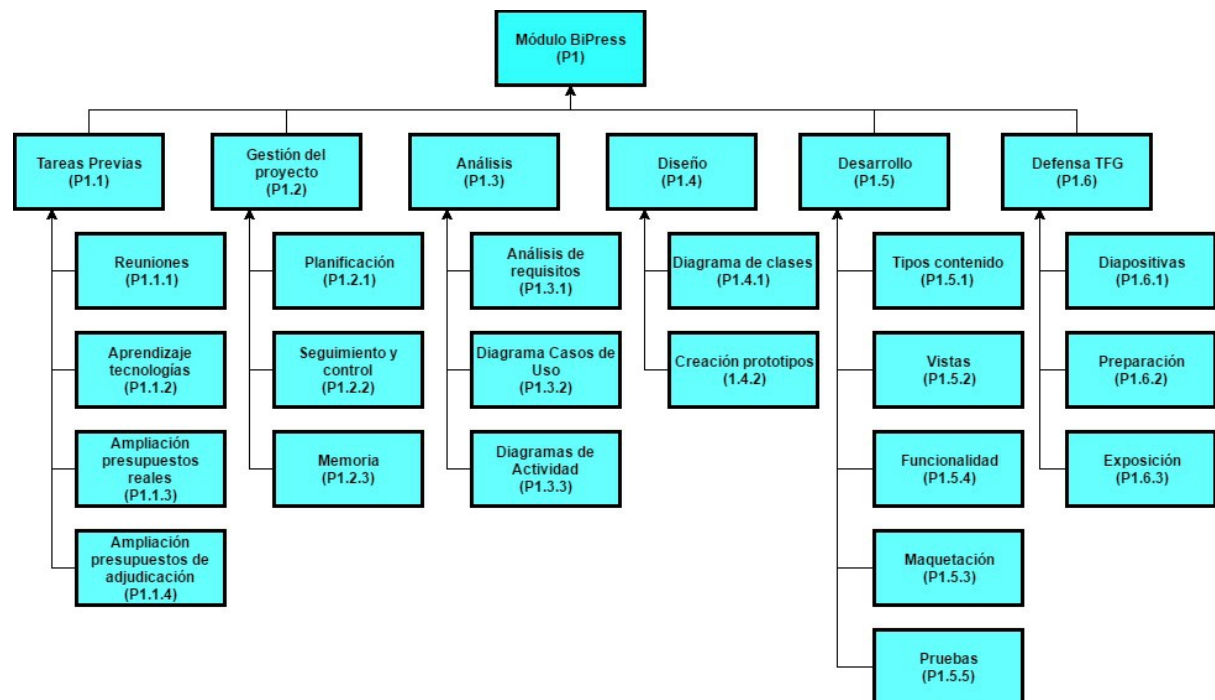


Diagrama 2.1: Diagrama EDT con las tareas del proyecto

2.2. Diccionario del EDT

A continuación se muestra un listado con todas las hojas del anterior EDT explicados para su correcta comprensión:

P1 Módulo BiPress: Se trata del proyecto en general.

P1.1 Tareas Previas: Contiene todas las tareas previas que se han tenido que realizar antes de empezar con el proyecto.

P1.1.1 Reuniones: Comprende todas las reuniones que se han tenido que realizar para la correcta comprensión del proyecto..

P1.1.2 Aprendizaje tecnologías: Se corresponde con el tiempo empleado tanto en el estudio de alternativas como en el estudio de las tecnologías a utilizar. En este caso, buena parte del aprendizaje se consideró como parte del periodo de prácticas, en concreto lo estudiado sobre Drupal y PHP. Por tanto, no se contabiliza como dedicación al proyecto.

P1.1.3 Ampliación presupuestos reales: Se trata de las tareas necesarias para acabar la infraestructura utilizada en la gestión de los presupuestos reales (partes de trabajo de obras). Esta tarea es una continuación del trabajo realizado durante el periodo de prácticas en la empresa, y que será explicado más adelante.

P1.1.4 Ampliación presupuestos de adjudicación: Tareas necesarias para acabar la infraestructura utilizada para la gestión de los presupuestos de adjudicación.

P1.2 Gestión del proyecto: Contiene todas las tareas necesarias para poder llevar a cabo la gestión del proyecto tales como la planificación, el seguimiento y las tareas previas al comienzo del proyecto.

P1.2.1 Planificación: Documento entregable que formará parte de la memoria final del proyecto y que contiene información sobre la organización llevada a cabo para realizar el trabajo satisfactoriamente.

P1.2.2 Seguimiento y control: Documento entregable que formará parte de la memoria final del proyecto y que contiene información sobre todo lo que ocurra a lo largo de la creación de dicho proyecto.

P1.2.3 Memoria: Se trata del documento entregable final que contiene toda la información relativa al proyecto.

P1.3 Análisis: Comprende el estudio de la funcionalidad del proyecto.

P1.3.1 Análisis de requisitos: Comprende el estudio de requisitos tanto funcionales como no funcionales.

P1.3.2 Diagrama Casos de Uso: Contiene el diagrama de casos de uso del proyecto.

P1.3.3 Diagramas de Actividad: Contiene varios diagramas de actividad que nos ayudan a saber cuál va a ser el funcionamiento de la aplicación.

P1.6 Diseño: Comprende el diseño que va a tener el módulo, incluyendo prototipos de la interfaz.

P1.6.1 Diagrama de clases: Diagrama que mostrará las clases que están implicadas en el proyecto.

P1.6.2 Creación prototipos: Contiene unas capturas con los prototipos del módulo.

P1.5 Desarrollo: Comprende todos los procesos de implementación del producto final.

P1.5.1 Tipos contenido: Proceso en el que se crean todos los campos necesarios (inputs, desplegables, formularios, etc) para poder crear la vistas necesarias del trabajo desde Drupal.

P1.5.2 Vistas: Es otro proceso de Drupal en el cual estamos creando las vistas necesarias del trabajo utilizando los tipos de contenido creados en el anterior punto.

P1.5.3 Maquetación: Proceso en el que se aplican los estilos necesarios para que las vistas creadas en el anterior apartado tengan el mismo aspecto que el del diseño.

P1.5.4 Funcionalidad: Es un proceso en el que se añade funcionalidad extra a ciertos campos para conseguir el comportamiento deseado (uso de javascript, peticiones AJAX, etc).

P1.5.5 Pruebas: Fase en la cual se realiza todo tipo de pruebas en cada una de las funcionalidades del módulo creado.

P1.6 Defensa TFG: Engloba todo el proceso seguido desde la creación de la presentación hasta su exposición ante el tribunal.

P1.6.1 Diapositivas: Fase en la cual se crean las diapositivas de la presentación del TFG.

P1.6.2 Preparación: Estudio de la presentación creada anteriormente.

P1.6.3 Exposición: Exposición de la presentación ante el tribunal del TFG.

2.3. Tiempos estimados

A continuación se muestra una tabla con todas las tareas a realizar durante el proyecto:

Tareas	Horas estimadas
Tareas Previas	50 h
Reuniones	15 h
Aprendizaje tecnologías	5 h
Ampliación presupuestos reales	25 h

Ampliación presupuestos de adjudicación	5 h
Gestión del proyecto	63 h
Planificación	10 h
Seguimiento y control	12 h
Memoria	41 h
Análisis	7 h
Análisis de requisitos	3 h
Diagrama Casos de Uso	4 h
Diseño	12 h
Diagrama de clases	7 h
Creación prototipos	5 h
Desarrollo	163 h
Creación tipos de contenido	20 h
Preparación de vistas	25 h
Maquetación	13 h
Añadir funcionalidad	80 h
Pruebas	25 h
Defensa TFG	5 h
Total	300 h

Tabla 2.1. Tabla con las tareas del proyecto

2.4. Diagrama de Gantt

En la siguiente tabla se puede ver el diagrama de Gantt del proyecto, en el cual se puede observar el período de actividad de cada tarea realizada:

Tareas	marzo				abril				mayo				junio				Sem. 1	Sem. 2
	Sem. 1	Sem. 2	Sem. 3	Sem. 4	Sem. 1	Sem. 2	Sem. 3	Sem. 4	Sem. 1	Sem. 2	Sem. 3	Sem. 4	Sem. 1	Sem. 2	Sem. 3	Sem. 4		
Tareas Previas																		
Planificación																		
Seguimiento y control																		
Memoria																		
Análisis																		
Diseño																		
Creación tipos de contenido																		
Preparación de vistas																		
Maquetación																		
Añadir funcionalidad																		
Pruebas																		
Defensa TFG																		

Figura 2.1. Diagrama de Gantt

Como se observa en la anterior imagen, el comienzo del proyecto será el día 7 de marzo y se espera poder terminarlo para antes del 28 de mayo (sin tener en cuenta la finalización de la memoria y la defensa del trabajo). Para poder cubrir las 300 horas de trabajo holgadamente, se dedicarán unas 20 horas semanales (4 horas por día). También se ha decidido dividir el periodo de trabajo en 12 semanas para poder realizar el módulo, otras 3 semanas para poder terminar la memoria, y otras 2 semanas y media para preparar la defensa del trabajo.

2.5. Diagrama de Hitos

A continuación se muestra el diagrama de hitos del proyecto donde se pueden ver los hitos más importantes del proyecto con su fecha de finalización:

Tareas	marzo				abril				mayo				junio					
	Sem. 1	Sem. 2	Sem. 3	Sem. 4	Sem. 1	Sem. 2	Sem. 3	Sem. 4	Sem. 1	Sem. 2	Sem. 3	Sem. 4	Sem. 1	Sem. 2	Sem. 3	Sem. 4	Sem. 1	Sem. 2
Tareas Previas	Día 7 ◆																	
Planificación				Día 28 ◆														
Seguimiento y control												Día 28 ◆						
Memoria															Día 20 ◆			
Análisis		Día 10 ◆																
Diseño			Día 17 ◆															
Creación tipos de contenido					Día 6 ◆													
Preparación de vistas						Día 13 ◆												
Maquetación							Día 20 ◆											
Añadir funcionalidad											Día 19 ◆							
Pruebas												Día 26 ◆						
Defensa TFG																		Día 6 ◆

Figura 2.2. Diagrama de hitos

Capítulo 3. Análisis

En este capítulo se describirá el proceso de análisis del proyecto para tener una idea más concreta sobre cómo va a funcionar el módulo. En concreto se mostrará el diagrama de casos de uso, junto a la descripción de los requisitos funcionales y no funcionales.

3.1. Tareas previas

El estudiante estuvo realizando las asignaturas del primer semestre del curso 2016-2017, Prácticas Externas I y Prácticas Externas II en esta empresa. Durante este periodo de tiempo, trabajó en el desarrollo del sistema que controla la gestión de los Presupuestos Reales comentados anteriormente, creando para ello las siguientes vistas:

- **Vista general de partes de trabajo.** Esta vista muestra todos los partes de una obra agrupados por año y por mes. También permite la descarga de la información relativa a la obra en formato .pdf o .xls, permitiéndole al usuario filtrar la información que desee obtener (Información en un periodo determinado, de un capítulo en concreto, de una partida, etc).

The screenshot shows a web application interface for 'Control de costes'. At the top, there is a navigation bar with the text 'SOION / Área privada' and a user profile 'serafin1975'. A sidebar on the left contains a menu with options like 'Empresa', 'Proyectos', 'Contactos', 'Usuarios', 'Actividades', 'Documentos', 'Informes', 'Mensajes', 'Mi cuenta', and 'Salir'. Below the menu is a 'Filtros' section with input fields for 'Fecha inicio', 'Fecha final', 'Titulo', and dropdown menus for 'Capítulos' and 'Partidas'. The main content area is titled 'Control de costes' and features a red button 'CREAR PARTES DE TRABAJO AVANZADOS'. It displays a table of work parts grouped by month and year. The table has columns for 'Titulo', 'Fecha', and an action icon (a circle with a plus sign). The data is organized into sections for 'Año: 2017' and 'Año: 2016', with sub-sections for each month (Abril, Marzo, Febrero, Diciembre).

Año: 2017		
Mes: Abril		
Titulo	Fecha	
Parte trabajo 03/04/2017	Lunes, 3 Abril, 2017	+
Mes: Marzo		
Titulo	Fecha	
Parte de trabajo 14/03/2017	Martes, 14 Marzo, 2017	+
Parte de trabajo 06/03/2017	Lunes, 6 Marzo, 2017	+
Mes: Febrero		
Titulo	Fecha	
Parte de trabajo 28/02/2017	Martes, 28 Febrero, 2017	+
Parte de trabajo 15/02/2017	Miércoles, 15 Febrero, 2017	+
Parte de trabajo 07/02/2017	Martes, 7 Febrero, 2017	+
Año: 2016		
Mes: Diciembre		

Figura 1.1. Vista general Partes de Trabajo

- **Vista de creación de un parte de trabajo.** Esta vista nos permite crear la información que se muestra en un parte de trabajo en concreto, permitiendo añadir uno a uno cada trabajador, material o máquina.

SOION / Área privada

Crear Partes de trabajo avanzado

Agregar contenido > Crear Partes de trabajo avanzado

Empresa >

Proyectos >

Contactos >

Usuarios >

Actividades >

Documentos >

Informes >

Mensajes >

Mi cuenta >

Salir >

Título * Producción

Fecha

05/04/2017

Mano de obra

Trabajador * Capítulo * Partida Horas

Fontanería - Elagir una opción

Información avanzada

ELIMINAR

AÑADIR OTRO ELEMENTO

Maquinaria

Maquina * Capítulo * Partida Horas

Fontanería - Elagir una opción

Información avanzada

ELIMINAR

AÑADIR OTRO ELEMENTO

Materiales

Material * Capítulo * Partida Cantidad

Fontanería - Elagir una opción

Información avanzada

ELIMINAR

AÑADIR OTRO ELEMENTO

Figura 1.2. Vista de creación de Partes de Trabajo

- **Vista de edición de un parte de trabajo.** Esta vista es muy parecida a la anterior, pero la diferencia es que nos muestra toda la información que contiene un parte de trabajo y permite su modificación.

SOION / Área privada

Editar parte de trabajo avanzado

Parte trabajo 03/04/2017 > Editar

Parte trabajo 03/04/2017

Empresa >

Proyectos >

Contactos >

Usuarios >

Actividades >

Documentos >

Informes >

Mensajes >

Mi cuenta >

Salir >

Título * Producción

Parte trabajo 03/04/2017 150

Fecha

03/04/2017

Mano de obra

Trabajador * Capítulo * Partida Horas

Daniel Tuesta [nic] Fontanería 001F45 4,00

Información avanzada

ELIMINAR

Trabajador * Capítulo * Partida Horas

Daniel Tuesta [nic] ARMARIOS DE COCI AC001 4,00

Información avanzada

ELIMINAR

Trabajador * Capítulo * Partida Horas





Daniel Tuesta [nic] SEGURIDAD SE001 3,00

Información avanzada

ELIMINAR

Figura 1.3. Vista de edición de Partes de Trabajo

- **Vista de un parte de trabajo diario.** Esta vista nos muestra la información relativa a un parte de trabajo diario y, además, nos permite la descarga de dicha información en formato .pdf y .xls.

SOION / Área privada  serafin1975  6  

Parte de trabajo avanzado 21/12/2016 - 11:32

Parte de trabajo avanzado 21/12/2016 - 11:32

- Empresa >
- Proyectos >
- Contactos >
- Usuarios >
- Actividades >
- Documentos >
- Informes >
- Mensajes >
- Mi cuenta >
- Salir >

Proyecto: FASE I Con partes de trabajo avanzados

[VOLVER A PARTES DE TRABAJO AVANZADO](#)

Fecha: 21/12/2016

Mano de obra			
Trabajador	Capítulo	Partida	Horas
Daniel Tuesta	Electricidad	EL002TPC63	3.00
Sergio asdad	FONTANERÍA	001F45	2.00

Maquinaria			
Máquina	Capítulo	Partida	Horas
Maquina 1	FONTANERÍA	001F45	2.00
Maquina aiosdjasjd	Electricidad	EL002TPC63	1.00

Materiales			
Material	Capítulo	Partida	Cantidad
maeriall	Electricidad	EL002TPC63	1.00
maeriall	FONTANERÍA	001F45	1.00

[Descargar parte diario](#)
[Descargar parte diario](#)
[Descargar parte diario por capítulo](#)

Figura 1.4. Vista de un Parte de Trabajo Diario

3.2. Captura de requisitos

Nos centramos ahora en los aspectos propios del módulo BiPress.

El módulo a desarrollar debe permitir al usuario mostrar los presupuestos reales, de licitación y de adjudicación.

3.2.1. Requisitos funcionales

Presentamos un listado con los requisitos funcionales del sistema:

- El usuario puede seleccionar el tipo de presupuesto que desea mostrar.
- El usuario puede seleccionar el tipo de gráfica que desea mostrar, así como ciertas opciones de configuración de la gráfica.
- El usuario puede seleccionar el proyecto que desea mostrar.
- El usuario puede seleccionar el tipo de capítulo que desea mostrar.
- El usuario puede seleccionar el tipo de recurso (mano de obra, maquinaria y materiales) que desea mostrar, en caso de haber seleccionado el tipo de

presupuesto real.

- El usuario puede seleccionar el tipo de agrupación que desea realizar (por meses o por partes de trabajo), si ha elegido los presupuestos reales y una gráfica adecuada entre las consideradas para el caso solicitado.

3.2.2. Requisitos no funcionales

Los requisitos no funcionales del sistema son los siguientes:

Accesibilidad

- El módulo deberá funcionar correctamente en los siguiente navegadores:
 - Mozilla Firefox: versión 38 o superior.
 - Google Chrome: versión 43 o superior.
 - Internet Explorer: versión 9 o superior.
 - Microsoft Edge: versión 40 o superior.
- Debe ser multiplataforma, por lo que debe poder utilizarse como mínimo en Windows, en cualquier distribución de Linux y en Mac OS X.
- Además, debe poder utilizarse desde cualquier dispositivo (ordenador, smartphone, tablet, etc).

Eficiencia

- El sistema debe mostrar las gráficas en un tiempo inferior a 2 segundos.

Disponibilidad

- El sistema debe estar disponible las 24 horas del día, los 365 días del año.

Usabilidad

- El tiempo de aprendizaje del módulo debe ser inferior a una hora.
- El sistema elaborado debe mostrar mensajes de error explicativos para el usuario.

Seguridad

- Se realizarán copias de seguridad cada mes por la seguridad de la base de datos.

3.3. Diagrama de Casos de Uso

En este apartado se mostrará el diagrama de casos de uso del proyecto:

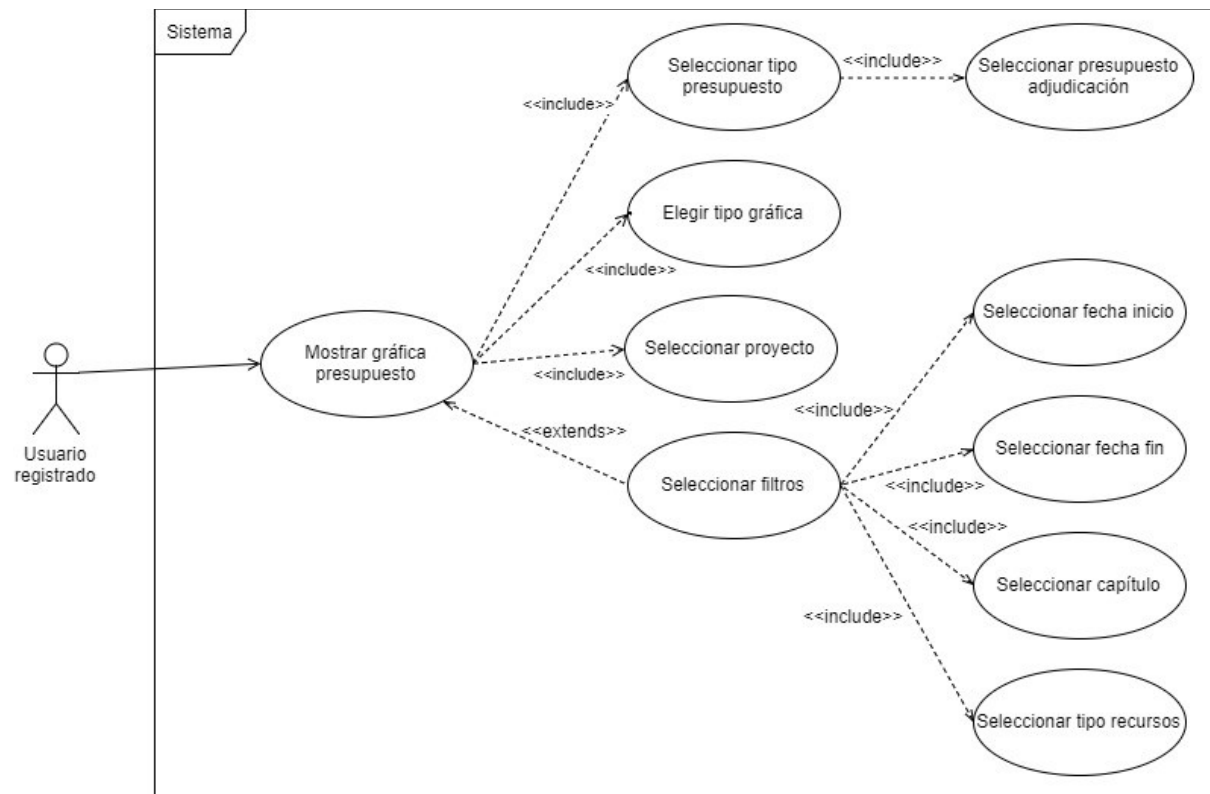


Figura 3.4.1. Diagrama de Casos de Uso

Para realizar el diagrama de casos de uso se ha utilizado la aplicación de diagramación draw.io, la cual está disponible de forma gratuita en Google Drive.

Respecto del anterior diagrama, indicar que todos los usuarios del módulo deberán ser usuarios registrados en el sistema (esta parte sale fuera de esta aplicación). El uso habitual es la solicitud de una gráfica de un tipo de presupuesto asociado a un proyecto, fijando algunas características de dicho solicitante.

Es necesario destacar que al trabajar con un CMS no ha sido necesario realizar el diseño de la base de datos ya que el propio CMS se encarga de gestionarla, por lo que en el siguiente capítulo no se hablará de esto. Lo que sí es necesario, es analizar el tipo de datos que se pueden registrar en la base de datos y cómo se va a acceder a los mismos.

En este caso, nos interesa saber que la base de datos contiene una tabla que contendrá 'nodos' (se corresponden con los tipos de contenido, sobre los cuales hablaremos más adelante), y a la cual necesitaremos acceder para obtener, por ejemplo, el nodo del proyecto para recuperar los capítulos y los partes de trabajo. También necesitaremos una tabla que contiene las colecciones de campo (hablaremos en el apartado de implementación sobre ellas), la cual necesitaremos para conseguir la información relativa a la mano de obra, maquinaria y materiales de un parte de trabajo.

Capítulo 4. Diseño

Este capítulo contiene el diseño que se ha realizado para el módulo a desarrollar. En concreto se mostrará un diagrama de clases que identificará las entidades que intervienen en el sistema, y también el diseño de interfaz, el cual nos mostrará cómo va a ser el diseño gráfico del módulo.

4.1. Diagrama de clases

A continuación se muestra el diagrama de clases del sistema:

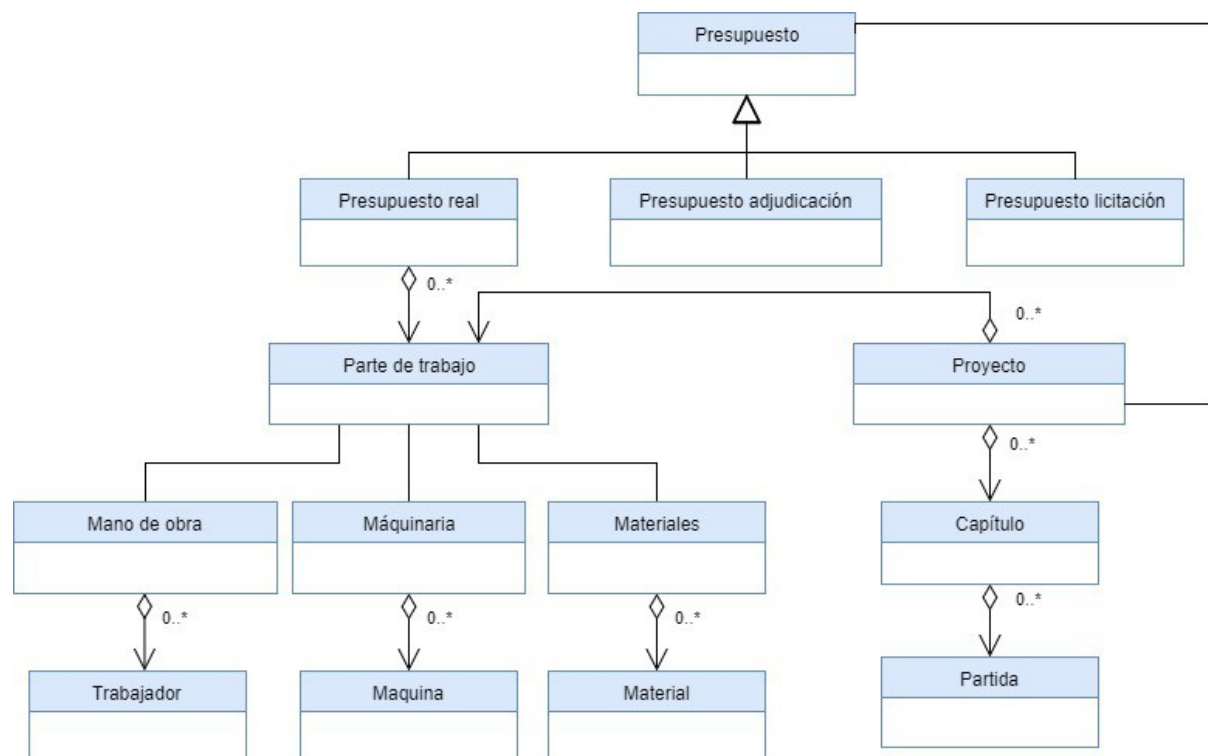


Figura 4.1. Diagrama de Clases

Una vez visto el diagrama de clases, conviene detallar algunos aspectos de cada una de sus clases:

- **Presupuesto:** Esta clase contiene un atributo de la clase 'Proyecto', necesario para obtener los presupuestos del proyecto seleccionado.
- **Presupuesto real:** Es una subclase de 'Presupuesto' y contiene el coste del presupuesto real, obtenido a partir de la información contenida dentro de los partes de trabajo.
- **Presupuesto adjudicación:** Es una subclase de 'Presupuesto' y contiene el coste del presupuesto de adjudicación.

- **Presupuesto licitación:** Es una subclase de 'Presupuesto' y contiene el coste del presupuesto de licitación.
- **Parte de trabajo:** Esta clase contiene demasiados atributos como para pintarlos todos, por lo que nombraremos sólo los utilizados para la realización del proyecto. Estos atributos son: el título del parte, el proyecto al que pertenece, la producción, la fecha de creación, la mano de obra, la maquinaria y los materiales.
- **Mano de obra:** Esta clase contiene únicamente un array con los trabajadores que la forman.
- **Maquinaria:** Esta clase contiene un array con las máquinas que la forman.
- **Materiales:** Esta clase contiene un array con los materiales que la forman.
- **Trabajador:** Los atributos utilizados de esta clase para el desarrollo del proyecto son los siguientes: el nombre del trabajador, los apellidos, su precio por hora, el capítulo en el que trabajan y la partida.
- **Maquina:** Los atributos utilizados de esta clase para el desarrollo del proyecto son los siguientes: el nombre de la máquina, su precio por hora, el capítulo en el que se ha utilizado y la partida.
- **Material:** Los atributos utilizados de esta clase para el desarrollo del proyecto son los siguientes: el nombre del material, su precio por unidad, el capítulo en el que se utilizó y la partida.
- **Proyecto:** De los atributos que contiene esta clase, los que se han utilizado son los siguientes: el id del proyecto, un array de capítulos y un array de partes de trabajo.
- **Capítulo:** De los atributos que contiene esta clase, los que se han utilizado son los siguientes: el id del capítulo, el proyecto al que pertenece, el título del capítulo y el array de partidas que contiene el capítulo.
- **Partida:** De los atributos que contiene esta clase, los que se han utilizado son los siguientes: el id de la partida, el capítulo al que pertenece, el título de la partida, la unidad de medición, la cantidad de recursos utilizados y el precio unitario de cada recurso.

Para realizar el diagrama de clases se ha utilizado draw.io, como en el caso del diagrama de casos de uso.

4.2. Diseño de la interfaz

En este apartado se expondrá el diseño de la interfaz de usuario utilizando para ello una serie de prototipos, los cuales han sido creados utilizando la herramienta de prototipado Balsamiq Mockups 3.

A la hora de realizar el diseño de la interfaz, se ha decidido que el aspecto que ésta debe tener ha de ser muy similar al del resto de la aplicación. En primer lugar se explican cada uno de los bloques que aparecen en el diseño.

- **Menú superior.** En la parte superior de la ventana aparecerá un menú que nos permitirá acceder a la cuenta de nuestro usuario, consultar las notificaciones y la mensajería, y nos dará también la opción de desplegar otro submenú que contendrá las mismas opciones que el menú lateral izquierdo (a excepción de los filtros).
- **Bloque selección.** Este bloque aparecerá justo debajo del menú anterior, y su objetivo es mostrar el nombre de la opción seleccionada del menú lateral izquierdo.
- **Menú lateral izquierdo.** Este menú permitirá movernos a través de la aplicación sin ninguna dificultad. Ofrecerá la opción de dirigirnos a la información de nuestra empresa, consultar los proyectos, dirigirnos a las gráficas y salir de la aplicación web.
- **Bloque de filtros.** Este bloque se colocará justo debajo del menú anterior y contendrá todos los filtros necesarios para mostrar las gráficas que deseemos.
- **Bloque de resultados.** Este bloque se encontrará justo en el centro de la ventana y contendrá los resultados de la vista. Si no se han rellenado los campos obligatorios, el bloque de resultados nos mostrará un mensaje de error indicando los campos que son necesarios rellenar. Si por el contrario se han rellenado los campos obligatorios, el bloque mostrará la gráfica correspondiente junto a un resumen de presupuestos que se encontrará justo debajo de la gráfica.

A continuación se muestra los prototipos utilizados:

Página de inicio

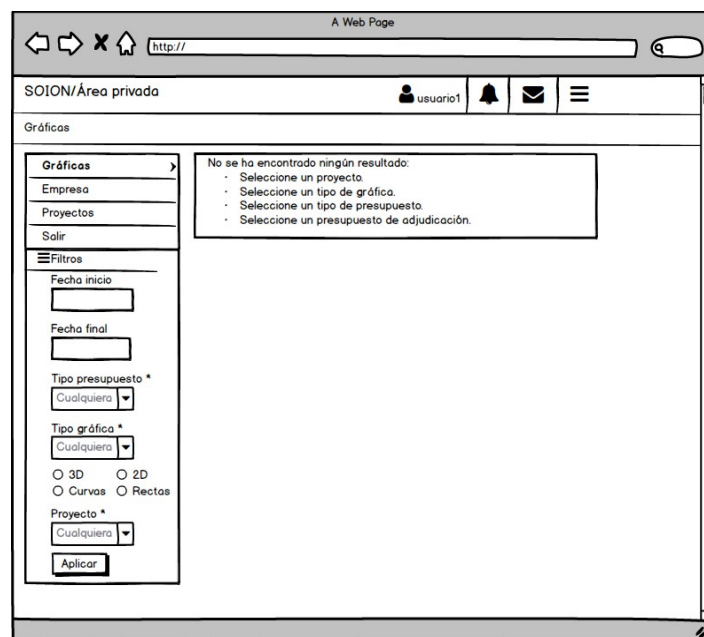


Figura 4.2.1. Página de inicio

Página presupuesto real (Pie Chart)

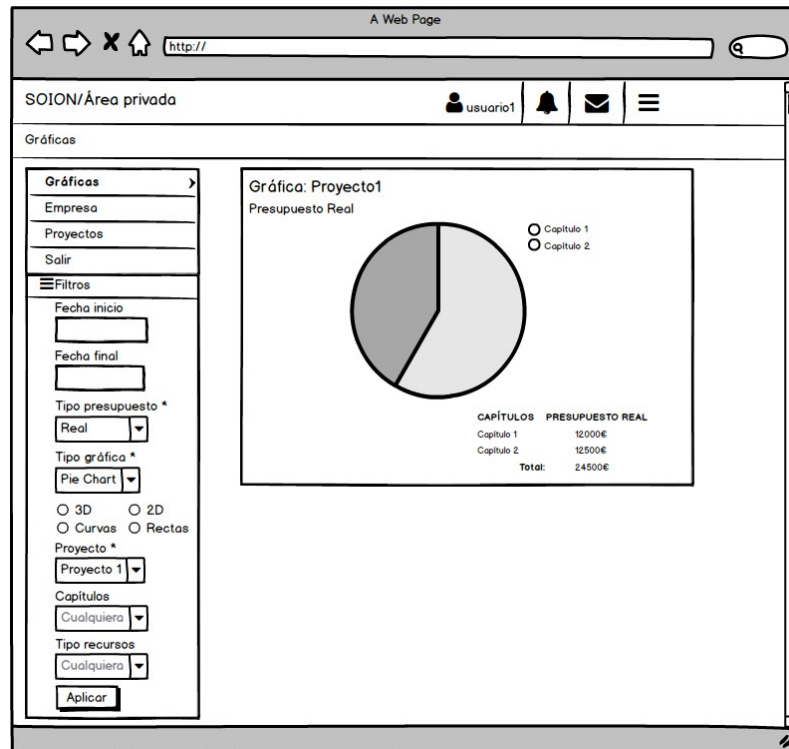


Figura 4.2.2. Página presupuesto real (Pie Chart)

Página presupuesto real (Column Chart)

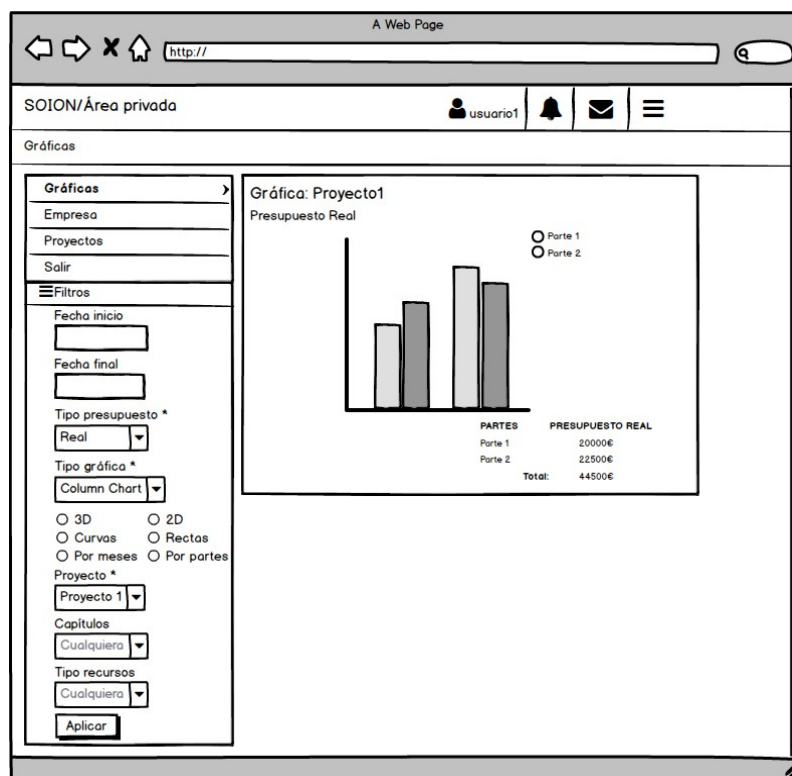


Figura 4.2.3. Página presupuesto real (Column Chart)

Página presupuesto de licitación

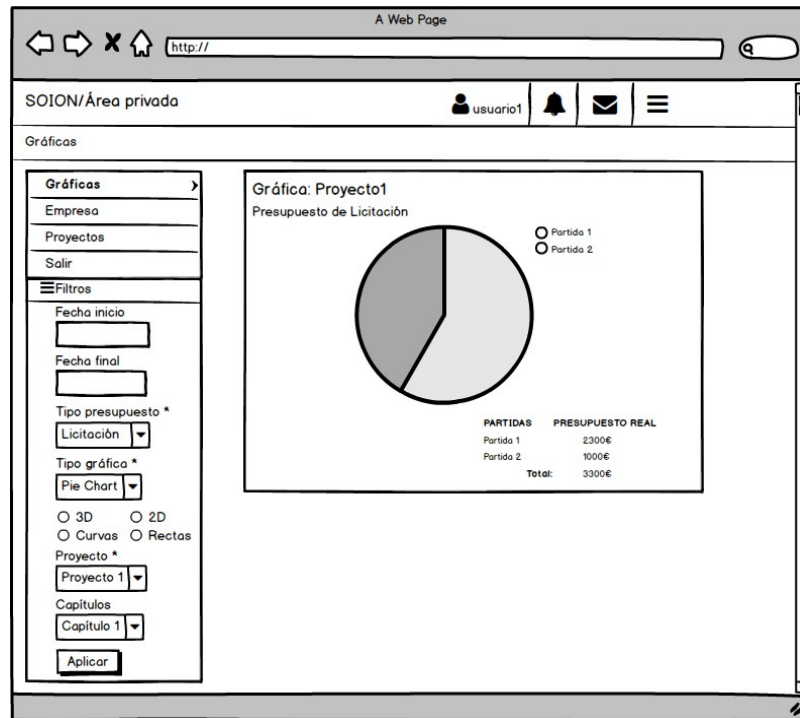


Figura 4.2.4. Página presupuesto de licitación

Página presupuesto adjudicación

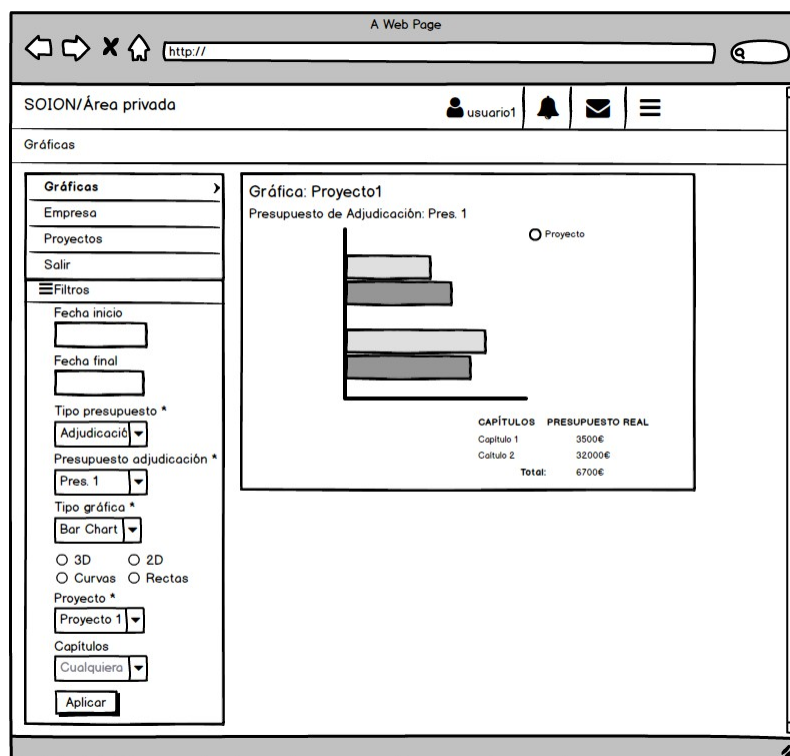


Figura 4.2.5. Página presupuesto adjudicación

Una vez vistos los prototipos es necesario realizar una serie de observaciones:

- En la página de inicio, una vez que rellenamos los campos obligatorios (en el campo 'Tipo presupuesto' tiene que estar seleccionada la opción 'Real') y pulsamos en el botón 'Aplicar' nos mostrará una ventana como la que aparece en la segunda figura (en este caso nos mostrará un Pie Chart). Como vemos, en esta ventana se incluirán dos nuevos campos: Capítulos y Tipo recursos. El campo 'Capítulos' nos mostrará todos los capítulos que tiene asociado el proyecto seleccionado, y el campo 'Tipo recursos' nos permitirá elegir si queremos mostrar el presupuesto de la mano de obra, la maquinaria y los materiales.
- Si elegimos la opción 'Real' del tipo de presupuesto y un tipo de gráfica que no sea la 'Pie Chart', nos mostrará una ventana como la de la tercera figura. La única diferencia que hay con la anterior gráfica es la inclusión de un nuevo 'radio button' que nos permitirán elegir el tipo de agrupación que se va a realizar (Si los presupuestos se van a agrupar por meses o por partes de trabajo). Estos 'radio button' serán necesarios ya que los presupuestos reales obtienen los costes a través de los partes de trabajo, los cuales siguen una línea temporal y por eso nos permite agrupar los presupuestos.
- En el caso de los presupuestos de adjudicación, se necesitará rellenar otro campo obligatorio llamado 'Presupuesto adjudicación'. Este campo nos mostrará los presupuestos de adjudicación que tiene el proyecto seleccionado. Este caso está representado en la última figura y como se puede observar, al elegir el tipo de gráfica 'Column Chart' ya no nos permitirá agrupar por meses o por partes de trabajo. Esto se debe a que la aplicación solo nos permitirá añadir un presupuesto de adjudicación que no será modificado.
- Con los presupuestos de licitación pasa lo mismo que en el anterior caso, pero con la diferencia de que no se añadirá el campo que nos permite elegir el tipo de presupuesto de adjudicación.

4.3. Aspecto final de la aplicación


En este apartado veremos cómo es el aspecto final de la interfaz de la aplicación.




Para empezar, indicar que para realizar el diseño visual de la aplicación se ha utilizado una plantilla de bootstrap, la cual nos viene con unos estilos por defecto para las fuentes, las tablas, etc, y que podemos elegir desde la ventana del administrador que nos ofrece Drupal.

A continuación veremos cómo han quedado las ventanas mostradas en el anterior apartado:

Página de inicio

SOION / Área privada

serafin1975



Gráficas

Gráficas >

Empresa >

Proyectos >

Salir >

Filtros

Fecha inicio

Fecha final

Tipo presupuesto *

- Cualquiera -

Tipo gráfica *

- Cualquiera -

☐ 3D ☒ 2D

☐ Curvas ☒ Rectas

Proyecto *

- Cualquiera -

APLICAR

No se ha encontrado ningún resultado:

- Seleccione un proyecto.
- Seleccione un tipo de gráfica.
- Seleccione un tipo de presupuesto.
- Seleccione un presupuesto de adjudicación.

Figura 4.3.1. Página de inicio

Página presupuesto real (Pie Chart)

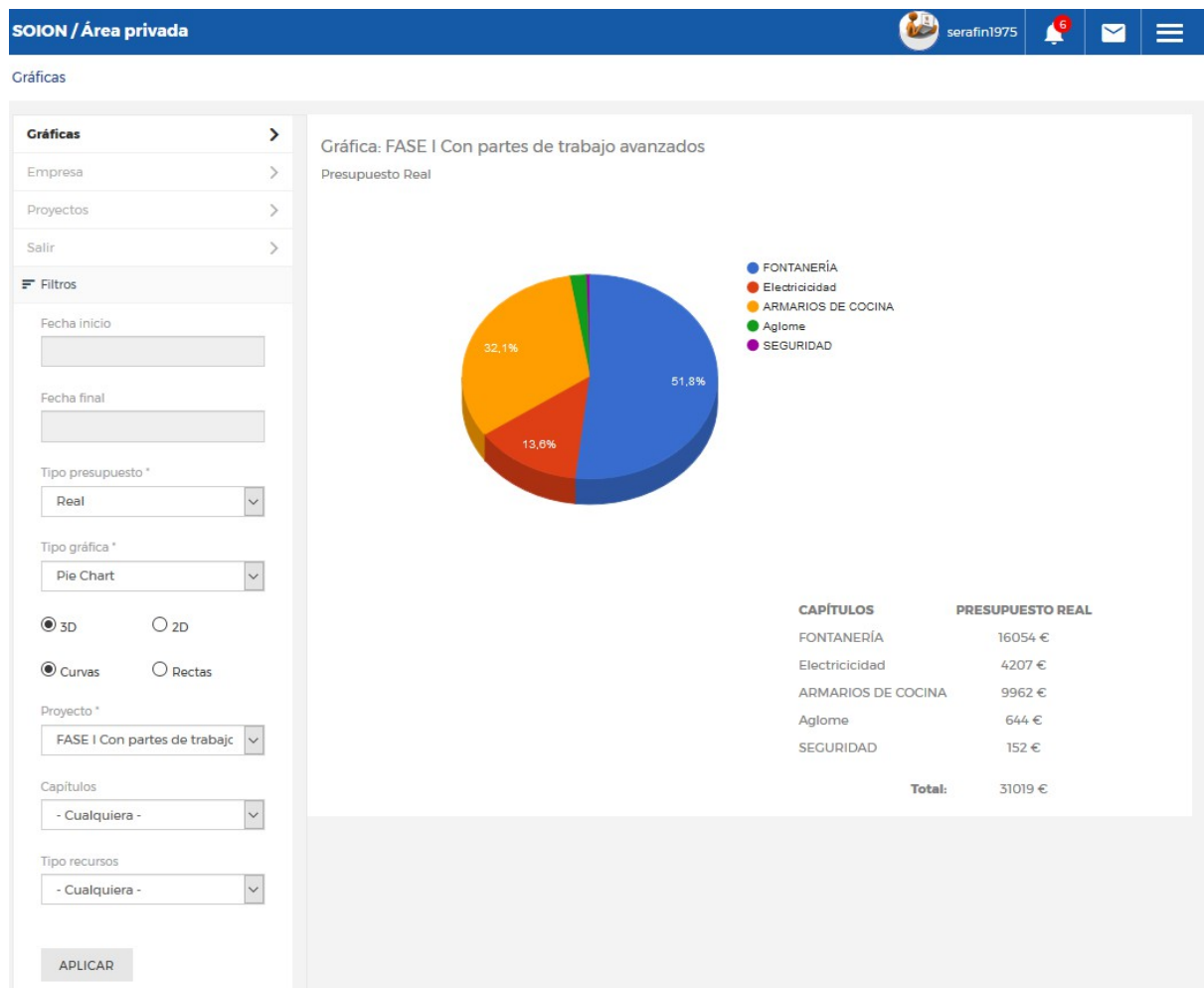


Figura 4.3.2. Página presupuesto real (Pie Chart)

Página presupuesto real (Column Chart)

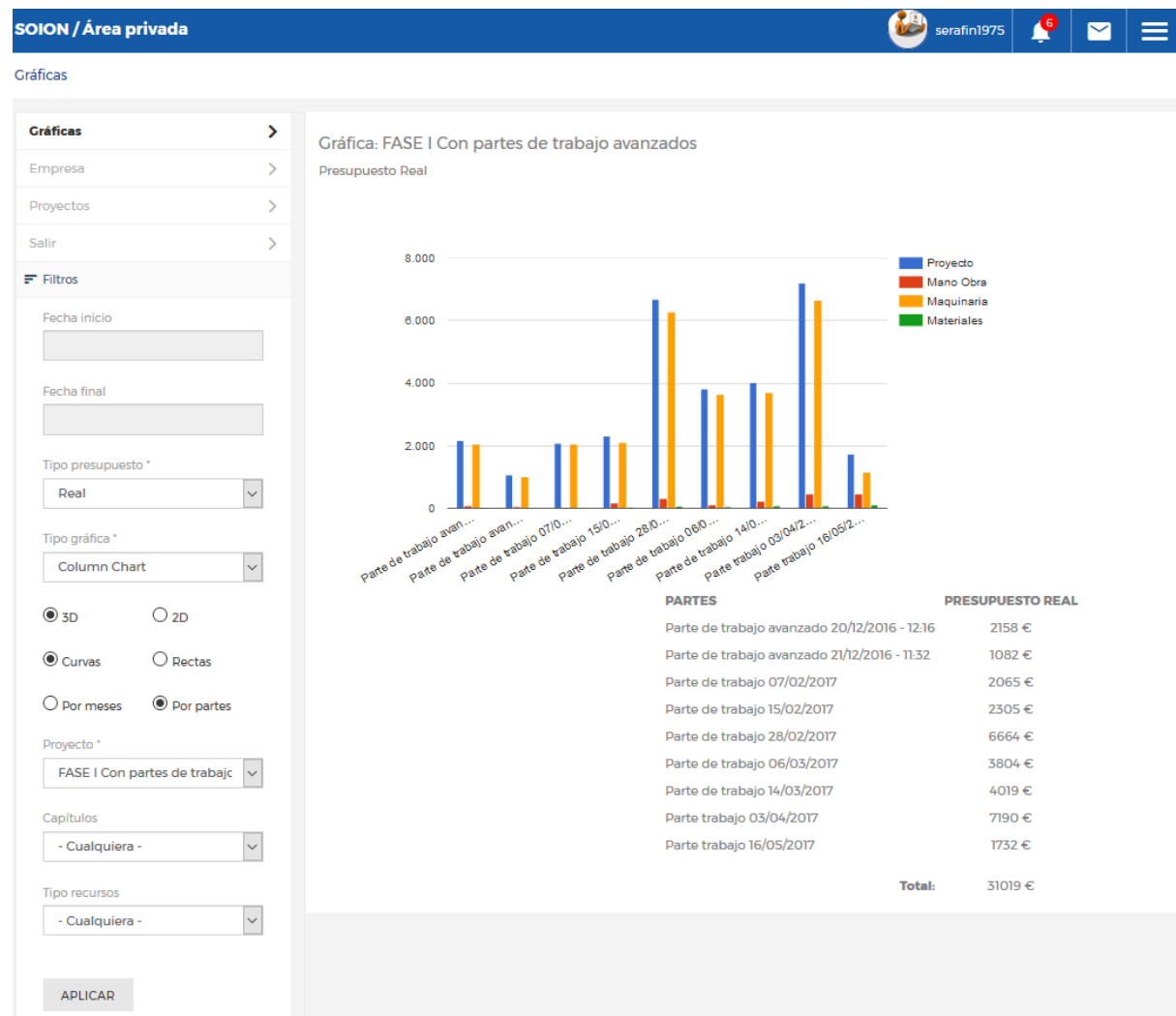


Figura 4.3.3. Página presupuesto real (Column Chart)

Página presupuesto de licitación

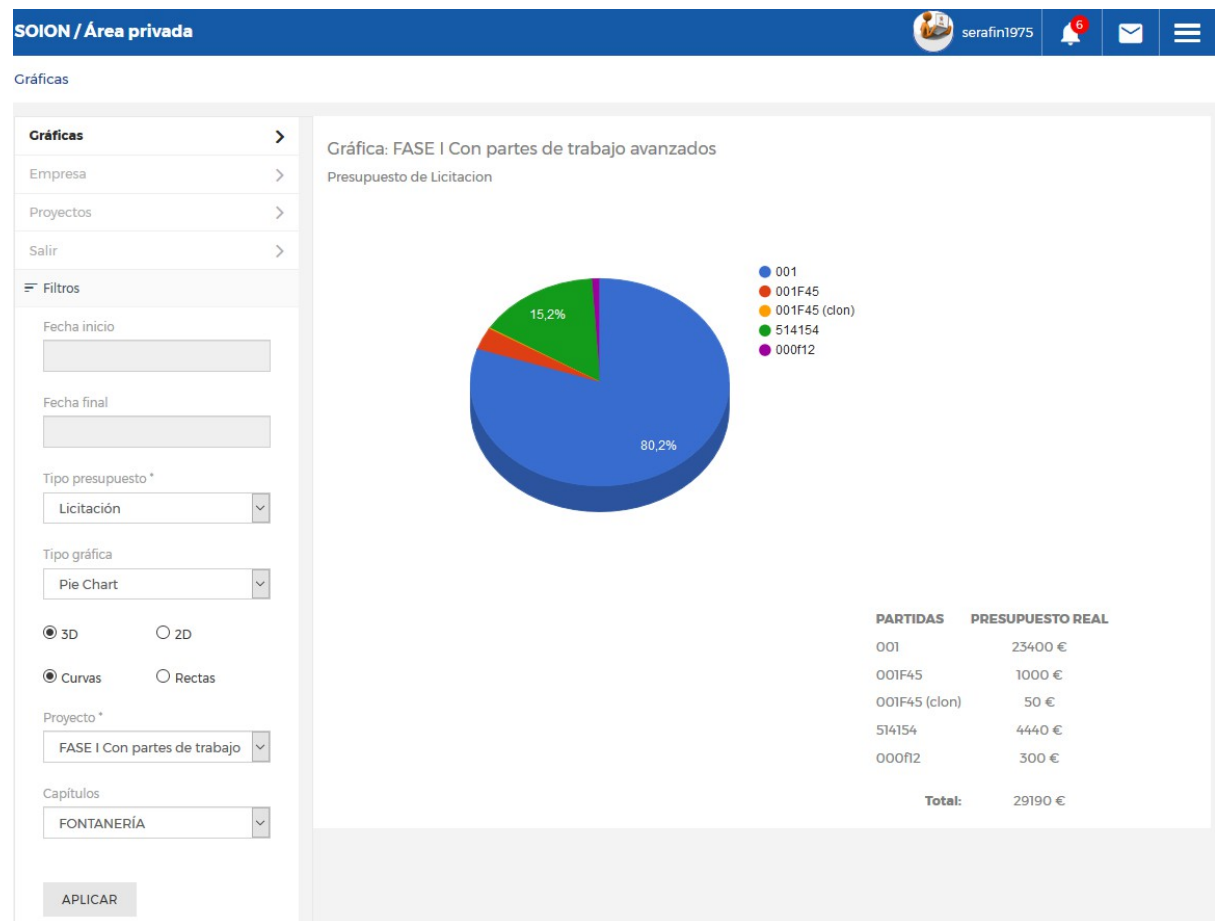


Figura 4.3.4. Página presupuesto de licitación

Página presupuesto adjudicación

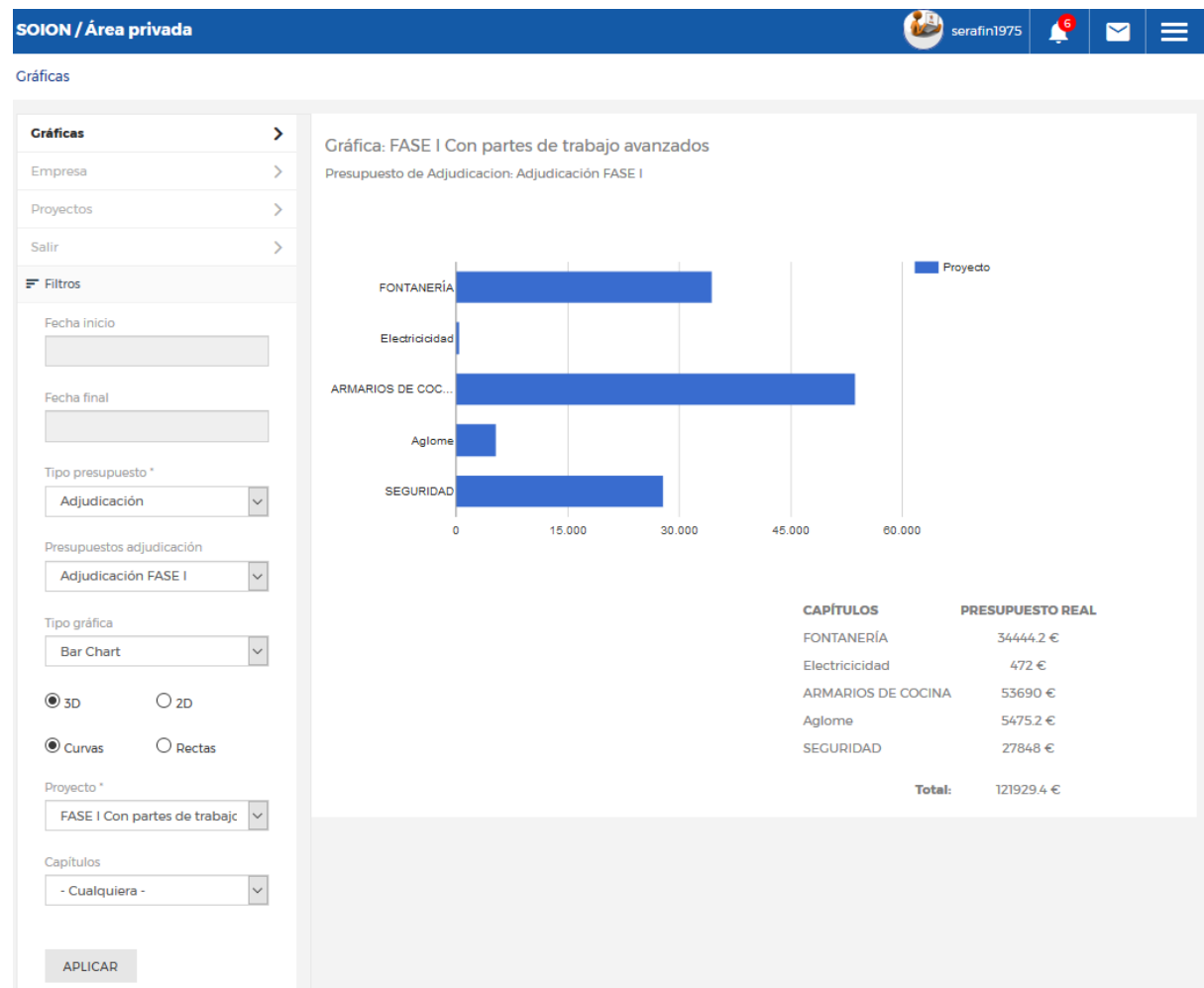


Figura 4.3.5. Página presupuesto adjudicación

Se puede observar el resultado final se parece mucho a los prototipos creados anteriormente.

Capítulo 5. Implementación

En este capítulo explicaremos cuál ha sido el proceso que se ha seguido para desarrollar el proyecto. Remarcar que el proyecto se ha desarrollado usando Drupal. El aprendizaje de esta herramienta se realizó durante el periodo de prácticas entre septiembre y diciembre de 2016.

Drupal permite modificar una aplicación web desde la ventana del administrador y, opcionalmente, desde 'back' (mediante el uso de ficheros con código que permiten añadir nuevas funcionalidades a la aplicación) utilizando un programa FTP para subir archivos con código al servidor. La segunda opción nos permite añadir funcionalidad extra a la web, lo cual hace de Drupal un CMS bastante completo y complejo a la vez.

En los siguientes apartados se explicará el proceso seguido para realizar la aplicación web utilizando Drupal. Para empezar, debemos utilizar la ventana del administrador que nos ofrece Drupal para crear toda la infraestructura necesaria para desarrollar la aplicación. Finalmente se hará uso del 'back' para añadir la funcionalidad necesaria.

5.1. Implementación usando el administrador

En este apartado se mostrará la infraestructura creada con la ventana del administrador y las opciones utilizadas para crearla.

Podemos observar que la ventana de administración cuenta con un menú en el que se pueden distinguir varias opciones. En nuestro caso, para crear la infraestructura solo ha sido necesario acceder a las opciones 'Estructura' y 'Módulos'.

A continuación veremos qué funcionalidad nos ofrecen las opciones citadas anteriormente:



Figura 5.1.1. Funcionalidad Estructura

De las funcionalidades de 'estructura' se han utilizado las siguientes:

- Bloques.
- Colecciones de campo.
- Menús.
- Tipos de contenido.
- Vistas.

La secuencia habitual de desarrollo comienza por crear los tipos de contenido, que se utilizan para crear las distintas entidades que van a intervenir en el sistema. Permiten la introducción de nuevas entidades en la aplicación.

A continuación se muestra cómo se ha creado el tipo de contenido 'Partes de trabajo avanzado':

Partes de trabajo avanzado				
EDITAR	CONTROL DE ACCESO	GESTIONAR CAMPOS	MANAGE DEPENDENCIES	GESTIONAR PRESENTACIÓN
CAMPOS DE COMENTARIOS COMMENT DEPENDENCIES				
Ordenar				
ETIQUETA	NOMBRE DE SISTEMA	TIPO DE CAMPO	CONTROL	OPERACIONES
+ Titulo	title	Elemento del módulo Node		
+ Opciones de ruta URL	path	Elementos de formulario del módulo Path		
+ Descripción	field_descripcion	Texto largo	Área de texto (varias filas)	editar eliminar States
+ Proyecto	field_proyecto	Referencia a nodo	Campo de texto de autocompletar	editar eliminar States
+ Producción	field_produccion	Texto	Campo de texto	editar eliminar States
+ Fecha	field_fecha	Fecha	Calendario desplegable	editar eliminar States
+ Año	field_ano	Número entero	Campo de texto	editar eliminar States
+ Mes	field_mes	Texto	Campo de texto	editar eliminar States
+ Mano de obra	field_mano_de_obra	Colección de campo	Incrustado	editar eliminar States
+ Maquinaria	field_maquinaria	Colección de campo	Incrustado	editar eliminar States
+ Materiales	field_materiales	Colección de campo	Incrustado	editar eliminar States
+ Ubicación	locations	Location module form elements		

Figura 5.1.2. Tipo de contenido 'Partes de trabajo avanzado'

Se puede observar que tiene una serie de campos entre los cuales aparece el 'Proyecto', que será otro tipo de contenido con otros campos diferentes. También son campos de interés 'Mano de obra', 'Maquinarias' y 'Materiales', que como se indica en el tipo, son colecciones de campo.

Este será el aspecto que tiene la colección de campo 'Mano de obra':

Colección de campo field_mano_de_obra				
GESTIONAR CAMPOS				
MANAGE DEPENDENCIES				
GESTIONAR PRESENTACIÓN				
Ordenar				
ETIQUETA	NOMBRE DE SISTEMA	TIPO DE CAMPO	CONTROL	OPERACIONES
+ Trabajador	field_trabajador	Referencia a nodo	Campo de texto de autocompletar	editar eliminar States
+ Capítulo	field_trabajos_realizados	Referencia a nodo	Lista de selección	editar eliminar States
+ Partida	field_id_partida_obra	Referencia a nodo	Lista de selección	editar eliminar States
+ Horas	field_horas	Número decimal	Campo de texto	editar eliminar States
+ Información avanzada	group_informacion_avanzada	Conjunto de campos	fieldset collapsed required_fields sí classes group-informacion-avanzada field-group-fieldset	 eliminar
+ Descripción	field_descripcion	Texto largo	Área de texto (varias filas)	editar eliminar States
+ Imágenes	field_imagenes	Imagen	Multiupload	editar eliminar States

Figura 5.1.3. Colección de campo 'Mano de obra'

Esta colección de campos contiene tres referencias a nodo, con las cuales podremos obtener la información de los tipos de contenido 'Trabajador', 'Capítulo' y 'Partida'. De esta manera, en una vista podremos visualizar todos los campos del trabajador, el capítulo y la partida asociada a cada ítem de mano de obra.

El siguiente paso a realizar es la definición de las 'Vistas'. Las vistas corresponden con la capa de presentación, y se encargan de mostrar los campos que se especifiquen y con el formato deseado.

A continuación se verá el aspecto de la vista utilizada para crear el proyecto:

The screenshot shows the 'Detalles de Page' configuration page in Drupal. It includes sections for 'TÍTULO', 'FORMATO', 'CAMPOS', 'CRITERIOS DE FILTRADO', 'CRITERIO DE ORDENACIÓN', 'OPCIONES DE PÁGINA', 'ENCABEZADO', 'PIE DE PÁGINA', 'PAGINADOR', 'FILTROS CONTEXTUALES', 'RELACIONES', 'COMPORTAMIENTO SI NO HAY RESULTADOS', 'FORMULARIO EXPUESTO', and 'OTROS(S)'. Each section contains specific configuration options and buttons to add or modify settings.

Figura 5.1.4. Vista del proyecto

En la anterior vista aparecen apartados que requieren un pequeño análisis para llegar a comprender qué es lo que hace esta vista y, en general, la forma de trabajar con vistas en Drupal.

- **Formato:** Nos permite elegir el formato de salida de la información. Los formatos permitidos son los siguientes:

- ☐ Cuadrícula
- ☐ Editable table
- ☒ Google Chart Tools
- ☐ Lista HTML
- ☐ Lista sin formato
- ☐ Menú saltar
- ☐ Presentación de diapositivas
- ☐ Tabla
- ☐ Table with aggregation options
- ☐ Views Accordion
- ☐ Views Calc Table

También puede ajustar las [configuración](#) para el estilo seleccionado actualmente.

Figura 5.1.5. Formatos de salida

En nuestro caso hemos elegido el formato Google Chart Tools, que nos permite mostrar la información en una gráfica.

- **Campos:** Nos permite añadir los campos que deseemos que aparezcan en la vista. Además, nos permite incluirlos o excluirlos de la presentación de tal manera que si deseamos excluirlos de la presentación no aparecerán en esta, pero si hacemos un `'die(var_dump($view))'` estos campos sí aparecerán. La anterior instrucción lo que hace es detener el procesado de la página cuando se llegue a dicha instrucción y mostrar el valor de la variable introducida dentro del `var_dump`.
- **Criterios de filtrado:** Esta funcionalidad lo que hace es filtrar los datos que cumplan las condiciones de filtrado especificadas. En este caso, los criterios que se han tenido en cuenta han sido los siguientes:
 - Que estén publicados.
 - Que sean del tipo 'Partes de trabajo avanzado'.
 - Que la fecha de publicación esté entre las dadas.
 - Que estos partes de trabajo pertenezcan al proyecto seleccionado.
- **Criterio de ordenación:** Permite especificar el orden de publicación de los elementos. En este caso, se ha decidido que el contenido se publique según la fecha de publicación y en orden ascendente.
- **Opciones de página:** En este caso, lo interesante es observar que permite cambiar la ruta de la vista.
- **Relaciones:** En este caso no hemos utilizado relaciones, pero su funcionamiento es similar al del 'join' de las consultas sql.
- **Comportamiento si no hay resultados:** Permite especificar el comportamiento de la vista si no hay resultados.
- **Formulario expuesto:** Esta opción nos permite crear un bloque de filtros expuestos que le permitirán al usuario filtrar las opciones que desee. Es necesario indicar que los campos a mostrar en el bloque se indicarán desde la opción 'criterios de filtrado' marcando para ello la opción de 'expuesto'. Como se puede observar, en 'criterios de filtrado' hay algunos campos que aparecen con 'expuesto' entre paréntesis.
- **Otros:** Esta opción nos permite cambiar opciones bastante interesantes como el nombre del sistema, decidir si se va a utilizar AJAX y el tema que se va a utilizar para la vista. En este caso se ha decidido que el nombre del sistema sea 'page_1', que se utilice AJAX, y que el tema de la vista sea un tema de 'Bootstrap'.

Una vez definidas las vistas que interesan, lo siguiente es crear un nuevo menú (que aparecerá en el menú lateral izquierdo a lo largo de toda la navegación por la aplicación web).

Para ello tenemos que dirigirnos a la opción menú indicada anteriormente y crear un nuevo menú.

En este caso se ha creado un menú llamado 'Menú gráficas' y que tiene el siguiente aspecto en el administrador:

ENLACE DEL MENÚ	ACTIVADO	OPERACIONES
+ Gráficas	<input checked="" type="checkbox"/>	editar eliminar

Figura 5.1.6. Menú gráficas administrador

Se puede observar que el menú solamente contiene una opción, la cual tendrá como ruta de enlace la creada en la vista anterior (por lo que al hacer click en dicha opción, nos redireccionará a la vista citada).

Una vez creado este menú, tenemos que crear otro menú que contenga las opciones que deseemos incluir cuando accedamos a las gráficas.

El menú se ha llamado 'Menú secundario gráficas' y tiene el siguiente aspecto en el administrador:

ENLACE DEL MENÚ	ACTIVADO	OPERACIONES	
+ Empresa	<input checked="" type="checkbox"/>	editar	eliminar
+ Proyectos	<input checked="" type="checkbox"/>	editar	eliminar
+ Proyectos	<input checked="" type="checkbox"/>	editar	eliminar
+ Historial de envíos	<input checked="" type="checkbox"/>	editar	eliminar
+ Partidas Favoritas	<input checked="" type="checkbox"/>	editar	eliminar
+ Tipo Capitulo	<input checked="" type="checkbox"/>	editar	eliminar
+ Salir	<input checked="" type="checkbox"/>	editar	eliminar

Figura 5.1.7. Menú secundario gráficas

Este menú contendrá enlaces a la página principal de nuestra empresa, otro a la sección de proyecto y otro que nos permitirá salir de la aplicación.

El siguiente paso a realizar es incluir estos dos menús creados y el bloque de filtros en la opción 'Bloques', la cual nombramos anteriormente. Se trata de un proceso muy sencillo. En primer lugar buscamos los menús y el bloque creados, y les cambiamos el nombre de la región por 'Navegación' (este bloque es el que contiene el menú lateral izquierdo):

BLOQUE	REGIÓN	OPERACIONES
Navegación		
+ Menú soion	Navegación ▼	configurar
+ Menú Construpress	Navegación ▼	configurar
+ Menú graficas	Navegación ▼	configurar
+ menu-proyecto	Navegación ▼	configurar
+ menu-capitulo	Navegación ▼	configurar
+ User menu	Navegación ▼	configurar
+ Formulario expuesto: documentos-page	Navegación ▼	configurar
+ Formulario expuesto: documentos_bitacora-page	Navegación ▼	configurar
+ Formulario expuesto: filtro_mensajes-page	Navegación ▼	configurar
+ Formulario expuesto: partes_trabajo_avanzados-page	Navegación ▼	configurar
+ Formulario expuesto: partes_trabajo-page	Navegación ▼	configurar
+ Formulario expuesto: tareas-page	Navegación ▼	configurar
+ Menú secundario gráficas	Navegación ▼	configurar
+ Formulario expuesto: grafica_partes_de_trabajo_avanzados-page_1	Navegación ▼	configurar

Figura 5.1.8. Bloque navegación

El siguiente paso a realizar es cambiar la configuración de estos menús y del formulario expuesto. En este caso particular, las configuraciones se llevarán a cabo como sigue:

Título del bloque
Filtros

CSS class(es)
bloque_filtros

OPCIONES DE REGIÓN
Especificar en qué temas y regiones se muestra este bloque.

Soion (Bootstrap) (tema predeterminado)
Navegación ▼

Seven (tema de administración)
- Ninguno - ▼

Bootstrap
- Ninguno - ▼

Bartik
- Ninguno - ▼

Opciones de visibilidad

Páginas
Restringido a algunas páginas

Tipos de contenido
Sin restricción

Roles
Sin restricción

Usuarios
No personalizable

Mostrar el bloque en páginas específicas

☐ Todas las páginas excepto las que se enumeran

☒ Sólo las páginas enumeradas

☐ Páginas en las que este código PHP devuelva Verdadero (sólo para expertos)

gráficas
gráficas/*

Figura 5.1.9. Configuración formulario expuesto

Bloque 'Menu secundario gráficas'

Título del bloque
<none>

CSS class(es)

OPCIONES DE REGIÓN
Especificar en qué temas y regiones se muestra este bloque.

Soion (Bootstrap) (tema predeterminado)
Navegación ▼

Seven (tema de administración)
- Ninguno - ▼

Bootstrap
- Ninguno - ▼

Bartik
- Ninguno - ▼

Opciones de visibilidad

Páginas
Restringido a algunas páginas

Tipos de contenido
Sin restricción

Roles
Sin restricción

Usuarios
No personalizable

Mostrar el bloque en páginas específicas

☐ Todas las páginas excepto las que se enumeran

☒ Sólo las páginas enumeradas

☐ Páginas en las que este código PHP devuelva Verdadero (sólo para expertos)

gráficas
gráficas/*

Figura 5.1.10. Configuración menú secundario gráficas

Bloque 'Menú graficas'

Título del bloque

CSS class(es)

OPCIONES DE REGIÓN
 Especificar en qué temas y regiones se muestra este bloque.

Soion (Bootstrap) (tema predeterminado)

Seven (tema de administración)

Bootstrap

Bartik

Opciones de visibilidad

Páginas
Sin restricción

Tipos de contenido
Sin restricción

Roles
usuario autenticado

Usuarios
No personalizable

Mostrar el bloque en páginas específicas

☐ Todas las páginas excepto las que se enumeran
☐ Sólo las páginas enumeradas
☒ Páginas en las que este código PHP devuelva Verdadero (sólo para expertos)

Figura 5.1.11. Configuración menú gráficas

Como vemos, en todos los bloques podemos cambiar el título del bloque y añadir una clase CSS. Otra opción que es importante es la de añadir las páginas en las que se mostrará el bloque indicado.

Para finalizar este apartado, señalaremos los módulos que han sido necesarios en Drupal para poder llevar a cabo este proyecto. En este caso solamente ha sido necesario utilizar el módulo 'Google Charts Tools', que permite mostrar los datos considerados en las vistas en forma de gráficas, permitiendo configurar dichas gráficas.

A continuación se muestra una captura de pantalla del módulo dentro de Drupal:


▼ CHARTING				
ACTIVADO	NOMBRE	VERSIÓN	DESCRIPCIÓN	OPERACIONES
<input checked="" type="checkbox"/>	Google Chart Tools	7.x-1.4	Provide a Google Chart Tools API. Necesitado por: Site analytics dashboard (activado), Google Chart Tools Example (activado), Google Chart Tools Views Integration (activado)	
<input checked="" type="checkbox"/>	Google Chart Tools Example	7.x-1.4	Provide a Google Chart Tools Example. Necesita: Google Chart Tools (activado)	
<input checked="" type="checkbox"/>	Google Chart Tools Views Integration	7.x-1.4	Provides a Google Chart Tools Views style. Necesita: Google Chart Tools (activado), Views (activado), Chaos tools (activado)	
<input checked="" type="checkbox"/>	Site analytics dashboard	7.x-1.4	Site analytics dashboard. Necesita: Google Chart Tools (activado)	 Permisos

Figura 5.1.12. Módulo Google Charts Tools

Una vez realizadas estas acciones desde la ventana del administrador, necesitamos añadir más funcionalidad desde 'back'.

5.2. Implementación utilizando ficheros con código

En este apartado se verá la parte del 'back' de la aplicación, es decir, el código utilizado para poder añadir toda la funcionalidad al proyecto desarrollado.

Para la correcta comprensión del desarrollo de los dos primeros ficheros, se hace necesario explicar qué es un 'hook'. Un hook es una función php que permite a los módulos de Drupal una mayor interacción con el 'core'. De esta manera se pueden alterar los comportamientos que Drupal tiene por defecto, y agregar una nueva funcionalidad.

A continuación se mostrará alguna parte del contenido de los ficheros utilizados:

Fichero construpress.module

Dentro de este fichero, hemos tenido que añadir funcionalidad al hook 'construpress_views_pre_render', de forma que se modifique el resultado a mostrar en la vista:

```
/**
 * Implements Hook_views_pre_render().
 */
function construpress_views_pre_render(&$view)
```

Figura 5.2.1. Hook construpress_views_pre_render

Es importante destacar que para que el hook sea reconocido por el sistema, hay que poner delante el nombre de éste, en este caso será construpress. Se puede observar que este hook recibe solamente la vista a la que se accede, y con la que se va a trabajar dentro del hook.

Dentro de este hook, lo esencial es indicar el comportamiento que tendrá éste si el nombre de la vista es el que nos interesa. Esto se hace de la siguiente forma:

```
//Modificamos el resultado de la vista de las gráficas
if($view->name == 'grafica_partes_de_trabajo_avanzados'){
```

Figura 5.2.2. Nombre vista

Vemos que con la propiedad 'name' de la vista podemos obtener el nombre de dicha vista.

Lo que se hace después es obtener el valor de cada uno de los campos filtrados. Para ello basta lo siguiente:

```
//Recogemos los parámetros del filtro
$id_proyecto = $_GET['field_proyecto_nid'];
$id_capitulo = $_GET['field_id_capitulo_nid'];
$tipo_grafica = $_GET['field_tipo_grafica_nid'];
$tipo_presupuesto = $_GET['field_tipo_presupuesto_nid'];
$tipo_agrupacion = $_GET['field_agrupacion_nid'];
$tipo_recurso = $_GET['field_tipo_recursos_nid'];
$presupuesto_adjudicacion = $_GET['field_presupuestos_adjudicacion_nid'];
```

Figura 5.2.3. Recogida parámetros filtro

Una vez obtenidas estas variables, se trabajará con ellas para pintar las gráficas. Si no se reciben ni el proyecto, ni el tipo de presupuesto, ni el tipo de gráfica, entonces se devuelve un mensaje de error explicando los campos que faltan por rellenar. Si por el contrario se recibiesen estos campos, habría que obtener los presupuestos en función de estos

parámetros.

Fichero template.php

En este fichero, hemos tenido que añadir funcionalidad al hook 'soion_form_alter':

```
function soion_form_alter(&$form, &$form_state, $form_id)
```

Figura 5.2.4. Hook soion_form_alter

Este hook sigue la misma nomenclatura que el explicado anteriormente, aunque recibe tres parámetros: form (formulario a modificar), form_state (cada uno de los campos del formulario junto al valor seleccionado en cada uno de ellos), y form_id (id del formulario).

Este hook se utilizará únicamente para modificar el formulario que contiene los nuevos filtros utilizados. A continuación explicaremos brevemente su implementación.

En primer lugar, necesitamos localizar el formulario a modificar. Para ello, primero tendremos que poner una condición que sólo nos permita acceder a los formularios expuestos:

```
if($form_id == 'views_exposed_form'){
```

Figura 4.2.5. Condición formularios expuestos

Una vez dentro de los formularios expuestos, hay que localizar el formulario de la vista que queremos modificar, junto a la página que deseemos. Esto se realizará de la misma forma que en el anterior fichero, añadiendo una condición en la que se comprueba que el nombre de la vista es el deseado. Dentro de esta condición, se añaden cada uno de los filtros adicionales necesarios para permitir al usuario una mayor funcionalidad. Esto se consigue añadiendo al formulario (que es un objeto) los campos como si fueran arrays y organizados de la misma forma que los campos creados anteriormente en la ventana del administrador.

Fichero dewenir.js

Este fichero contendrá el código javascript necesario, junto a las peticiones AJAX realizadas. Utilizaremos el código javascript y las peticiones AJAX para controlar el comportamiento del tipo de gráficas. Dependiendo del tipo de presupuesto seleccionado, se mostrarán en el selector de tipos de gráficas unas opciones u otras. Para conseguir este comportamiento, ha sido necesario crear el siguiente código:


```

/* Controla el funcionamiento del select tipo gráfica de la vista de gráficas */
var funcionamientoTipoGrafica = {
  urlPeticion: window.location.protocol + "://" + window.location.host + '/privado/construpress/get_graficas_presupuesto/',
  init: function() {
    this.config();
    this.funcionamiento();
  },
  return;
},
config: function() {
  this.body = body;
  this.bloque_filtros = this.body.find('.leftLat .bloque_filtros');
  this.selectPresupuestos = this.bloque_filtros.find('#edit-field-tipo-presupuesto-nid');

  return;
},
funcionamiento: function() {
  var that = this;

  this.selectPresupuestos.on('change', function(e) {
    e.preventDefault();
    //Obtengo el elemento padre de cada campo de los filtros
    var elementoPadre = $(this).parent().parent().parent().parent();
    //Ahora obtengo el select de gráficas
    var selectGraficas = elementoPadre.find('#edit-field-tipo-grafica-nid');

    if(that.selectPresupuestos.val() == 'Real') {
      selectGraficas.append('<option value="AreaChart">Area Chart</option>');
      selectGraficas.append('<option value="LineChart">Line Chart</option>');
    }
    else
      that.peticionAjax($(this).val(), selectGraficas);
  });
},

```

Figura 4.2.7. Código javascript (I)

Este código lo que hace es crear una variable llamada ‘funcionamientoTipoGrafica’ que contiene una serie de funciones:

- `init()`: Función constructora que llamará al resto de funciones creadas.
- `config()`: Función de configuración que contiene las variables utilizadas. En este caso, contiene el `body`, el bloque de filtros y el selector de tipo de presupuestos.
- `funcionamiento()`: Esta función lo que hace es aplicar el manejador de eventos ‘onchange’ sobre el selector de tipo de presupuestos, de manera que cuando se cambie de tipo de presupuesto, cambien las opciones del tipo de gráfica. Si la opción seleccionada es ‘Real’, se procederá a añadir ‘Area Chart’ y ‘Line Chart’ a las opciones que vienen por defecto, si se selecciona cualquier otro presupuesto, se hará una petición AJAX como la siguiente:

```

peticionAjax: function(presupuesto, selectGraficas){
    var url = this.urlPeticion + presupuesto;console.log(url);
    var that = this;

    var jqxhr = $.ajax({
        url: url,
        type: "GET",
        cache: false,
        dataType: 'json'
    })
    .done(function(respuesta) {
        that.noMostrarGraficas(respuesta, selectGraficas);
        return;
    })
    .fail(function(){
        console.log("Error obtenido");
    });
    return;
},
noMostrarGraficas: function(graficas, selectGraficas){
    //Ocultamos los options que tengan la clave obtenida en la petición AJAX
    $.each(graficas, function(clave, valor){
        selectGraficas.find('option[value="' + clave + '"]').hide();
    });

    return;
}
}

```

Figura 4.2.8. Código javascript (II)

Cuando la respuesta sea correcta, se llamará a una función 'noMostrarGraficas' que se encargará de borrar todas las opciones del selector 'tipo gráficas' que tengan alguna clave que venga en la respuesta de la petición AJAX.

Es necesario mencionar que la petición AJAX se gestionará en el fichero 'construpress.module'. Dentro de este fichero se invocará a la función que devuelve el resultado de la petición. En concreto se realizará en el hook 'construpress_menu()':

```

/**
 * Implementación del hook_menu()
 */

function construpress_menu()

```

Figura 5.2.9. Hook construpress_menu

Dentro de este menú la idea es declarar un array de 'items', que contendrá el resultado de las peticiones AJAX que se realicen.

Una vez implementado este comportamiento, lo único que tenemos que hacer es invocar a la variable creada anteriormente(funcionamientoTipoGrafica) dentro de una función principal, utilizando para ello la función init() creada dentro de la variable:

```

if(body.hasClass('page-graficas')){
    funcionamientoTipoGrafica.init();
}

```

Figura 4.2.10. Código javascript (III)

Se puede ver en la anterior captura que esta función la invocamos si estamos en la página que nos interesa, en este paso 'page-graficas'.

Fichero views-view--grafica-partes-de-trabajo-avanzados.tpl.php

Este fichero tiene el contenido de la vista que se especifica en el nombre del fichero, en este caso se trata de la vista grafica-partes-de-trabajo-avanzados. Es importante citar que este fichero se puede generar desde la ventana del administrador de Drupal, dirigiéndose al apartado 'Vistas' y seleccionando la vista que queremos. Una vez dentro, tenemos que dirigirnos a la opción 'Tema' y pulsar en la opción 'Salida de presentación' para que nos muestre la estructura HTML del fichero.

Se ha utilizado para modificar el contenido de la vista. En concreto, se han añadido un título y un subtítulo a la vista, y una tabla debajo de la gráfica con el resumen de los resultados obtenidos.

A continuación se muestran capturas de pantalla del contenido del fichero junto a algunas explicaciones:

```
<?php
//Obtenemos las opciones seleccionadas en el bloque de filtros
$idproyecto = $view->exposed_raw_input['field_proyecto_nid'];
$tipo_grafica = $view->exposed_raw_input['field_tipo_grafica_nid'];
$tipo_recursos = $view->exposed_raw_input['field_tipo_recursos_nid'];
$tipo_presupuesto = $view->exposed_raw_input['field_tipo_presupuesto_nid'];
$capitulo = $view->exposed_raw_input['field_id_capitulo_nid'];
$tipo_agrupacion = $view->exposed_raw_input['field_agrupacion_nid'];
$presupuesto_adjudicacion_nid = $view->exposed_raw_input['field_presupuestos_adjudicacion_nid'];
//Cargamos el nodo del presupuesto de adjudicación seleccionado
$presupuesto_adjudicacion = node_load($presupuesto_adjudicacion_nid);
//Cargamos el nodo del proyecto seleccionado
$proyecto = node_load($idproyecto);
//Obtenemos el título del proyecto
$title = 'Gráfica: '.$proyecto->title;

if($tipo_presupuesto == 'Real'){
    $subtitulo = 'Presupuesto '.$tipo_presupuesto;
}
if($tipo_presupuesto == 'Licitacion'){
    $subtitulo = 'Presupuesto de '.$tipo_presupuesto;
}
if($tipo_presupuesto == 'Adjudicacion'){
    $subtitulo = 'Presupuesto de '.$tipo_presupuesto.': '.$presupuesto_adjudicacion->title;
}
```

Figura 4.2.11. Contenido views-view--grafica-partes-de-trabajo-avanzados.tpl.php (I)

En la anterior captura se puede observar que, en primer lugar, se obtienen los valores de los campos seleccionados en el bloque de filtros, y se crean el título y el subtítulo de la vista.

Una vez obtenidos los valores anteriores, tendremos que crear una tabla nueva (que contendrá el resumen de la gráfica) utilizando los valores obtenidos de la gráfica, justo debajo de ésta.

Fichero style.css

Este fichero contiene los estilos utilizados en la vista del proyecto. En este caso no ha sido necesario realizar demasiadas modificaciones en la plantilla de bootstrap seleccionada, por lo que este fichero se ha utilizado muy poco.

El contenido de este fichero es el siguiente:

```
/* Página de comparación de gráficas */  
  
.page-graficas .main-container .form-radios > div { width: 50%; display: inline-block; }  
.page-graficas .main-container .views-exposed-form .views-submit-button { display: inline-block; }  
  
.page-graficas .main-container table { float: right; margin: 0 90px; }  
.page-graficas .main-container table th { padding: 5px; }  
.page-graficas .main-container table tr td { padding: 5px; }  
.page-graficas .main-container table tr #total { padding-top: 20px; font-weight: bold; float: right; }  
.page-graficas .main-container table tr #precio { padding-top: 20px; }  
.page-graficas .main-container table tr td:last-of-type { text-align: center; }
```

4.2.13. Contenido style.css

Únicamente se ha modificado el tamaño y la disposición de los radio button del formulario expuesto, la disposición del botón 'Aplicar' del bloque de filtros y los estilos de la tabla que contiene el resumen de cada gráfica.

Capítulo 6. Pruebas

En este capítulo se mostrarán las pruebas que se han realizado con la aplicación para detectar posibles errores de funcionalidad, de diseño, etc. Se señalan también los resultados obtenidos en cada una de ellas.

6.1. Identificación de las pruebas a realizar

En la siguiente tabla se muestran las pruebas realizadas con una descripción de cada una de ellas:

Pruebas	Descripción
Comprobar funcionamiento en diferentes navegadores.	Utilizar la aplicación en diferentes navegadores.
Comprobar que se ve correctamente en los diferentes navegadores.	Utilizar la aplicación en diferentes navegadores.
Comprobar funcionamiento del bloque de filtros.	Comprobar que una vez aplicados los filtros, los resultados obtenidos por el módulo son correctos.
Comprobar funcionamiento del javascript utilizado.	Comprobar que si se selecciona la opción 'Real' en el tipo de presupuesto, se añaden las opciones 'Area Chart' y 'Line Chart' al select 'tipo de gráficas'. Comprobar que si se seleccionan la opción 'Licitación' o 'Adjudicación' en el tipo de presupuesto, se realiza correctamente la petición AJAX desarrollada.
Comprobar que las gráficas se muestran para todos los tipos de presupuesto.	Verificar que para cada tipo de presupuesto se muestran las gráficas seleccionadas sin ningún tipo de error.
Comprobar datos del resumen.	Comprobar que los datos que aparecen en las gráficas coinciden con los datos del resumen final.

Tabla 6.1. Pruebas a realizar y descripción

6.2. Resultado de las pruebas realizadas

En la siguiente tabla se muestran las pruebas realizadas y los resultados obtenidos:

Pruebas	Resultado
Comprobar funcionamiento en diferentes navegadores.	Esta prueba fue realizada durante el periodo de desarrollo y de finalización de la aplicación. Se realizó en los siguientes navegadores: <ul style="list-style-type: none"> • Mozilla Firefox v53.0.3 • Google Chrome v58.0.3029.110 • Internet Explorer v11.0.42 • Microsoft Edge v38.14393.1066.0 El resultado fue satisfactorio, ya que todo el funcionamiento era correcto en todos los navegadores.

Comprobar que se ve correctamente en los diferentes navegadores.	<p>Esta prueba fue realizada durante el periodo de desarro y de finalización de la aplicación. Se realizó en los siguientes navegadores:</p> <ul style="list-style-type: none"> • Mozilla Firefox v53.0.3 • Google Chrome v58.0.3029.110 • Internet Explorer v11.0.42 • Microsoft Edge v38.14393.1066.0 <p>El resultado fue satisfactorio, ya que se veía correctamente en todos los navegadores y ningún control quedaba mal colocado.</p>
Comprobar funcionamiento del bloque de filtros.	<p>Esta prueba se realizó una vez finalizada la aplicación y en los navegadores nombrados en las anteriores pruebas.</p> <p>El resultado fue satisfactorio, ya que las gráficas mostraban correctamente los datos filtrados.</p>
Comprobar funcionamiento del javascript utilizado.	<p>Esta prueba se realizó una vez finalizada la creación del bloque de filtros, así como de su funcionamiento javascript. Se realizó en los navegadores nombrados en las anteriores pruebas.</p> <p>En un principio hubo algún error provocado por el comportamiento de los navegadores, pero pude arreglarlo fácilmente.</p> <p>Al final el resultado fue satisfactorio, ya que al seleccionar la opción 'Real' como tipo de presupuesto, las gráficas que aparecía eran 'Pie Chart', 'Column Chart', 'Bar Chart', 'Area Chart' y 'Line Chart'. Si se seleccionaba cualquiera de los otros dos presupuestos, las gráficas que aparecían eran 'Pie Chart', 'Column Chart' y 'Bar Chart'.</p>
Comprobar que las gráficas se muestran para todos los tipos de presupuesto.	<p>Esta prueba se realizó para cada tipo de presupuesto y una vez finalizada la aplicación. Se realizó en los navegadores nombrados en las anteriores pruebas.</p> <p>El resultado fue satisfactorio, ya que las gráficas mostraban correctamente los datos de cada tipo de presupuesto.</p>
Comprobar datos del resumen.	<p>Esta prueba se realizó una vez finalizada la creación del resumen. Se realizó en los navegadores nombrados en las anteriores pruebas.</p> <p>El resultado fue satisfactorio, ya que los datos que mostraba el resumen coincidían con los datos mostrados en las gráficas.</p>

Tabla 6.1. Pruebas a realizar y descripción

Capítulo 7. Seguimiento y control

En este apartado se va a realizar un seguimiento del proyecto controlando las horas empleadas, los problemas que ha habido durante la vida del proyecto y las modificaciones que han sido necesarias.

7.1. Tabla desviaciones

A continuación se muestra una tabla en la que se pueden observar las horas estimadas, las horas reales y la desviación de las tareas realizadas durante el proyecto:

Tareas	Horas estimadas	Horas reales	Desviación
Tareas Previas	50 h	60 h	+ 10 h
Reuniones	15 h	13.5 h	- 1.5 h
Aprendizaje tecnologías	5 h	6 h	+ 1 h
Ampliación presupuestos reales	25 h	37.5 h	+ 12.5 h
Ampliación presupuestos de adjudicación	5 h	3 h	- 2 h
Gestión del proyecto	63 h	64 h	+ 1 h
Planificación	10 h	7 h	- 3 h
Seguimiento y control	12 h	10 h	- 2 h
Memoria	41 h	47 h	+ 6 h
Análisis	7 h	5 h	- 2 h
Análisis de requisitos	3 h	2 h	- 1 h
Diagrama Casos de Uso	4 h	3 h	- 1 h
Diseño	12 h	15 h	+ 3 h
Diagrama de clases	7 h	8 h	+ 1 min
Creación prototipos	5 h	7 h	+ 2 h
Desarrollo	163 h	157 h	- 6 h
Creación tipos de contenido	20 h	10 h	- 10 h
Preparación de vistas	25 h	20 h	- 5 h
Maquetación	13 h	1 h	- 12 h

Añadir funcionalidad	80 h	94 h	+ 14 h
Pruebas	25 h	32 h	+ 7 h
Defensa TFG	5 h		
Total	300 h	301 h	+ 1 h

Tabla 7.1. Tabla de desviación de tiempo

7.2. Desviaciones significativas

Las desviaciones más significativas que ha habido a lo largo del proyecto han sido las siguientes:

- Como se puede observar en la tabla que se muestra en el anterior apartado, la primera desviación significativa que aparece se encuentra en las tareas previas. Esta desviación se ha producido debido a la ampliación de la infraestructura utilizada para los presupuestos reales (partes de trabajo), a la cual se han tenido que emplear 30 horas. Otra causa de esta desviación ha sido la terminación de la infraestructura utilizada para los presupuestos de adjudicación.
- En la gestión del proyecto no ha habido una desviación demasiado significativa en conjunto, aunque si nos fijamos en la desviación de la memoria vemos que ha habido una desviación de 6 horas debido a las modificaciones que se han tenido que hacer en la aplicación.
- En el desarrollo no ha habido una desviación global demasiado grande. Si nos fijamos en la desviación que ha habido en cada una de las tareas, podemos ver que ha habido desviaciones significativas:
 - **Creación de tipos de contenido:** Podemos ver que se ha reducido a la mitad el tiempo estimado con el real. Esta reducción simplemente se ha producido por un fallo de estimación, ya que se pensaba que iba a llevar más tiempo realizar esta tarea.
 - **Preparación de vistas:** En esta tarea ha habido una diferencia de 5 horas con respecto a lo estimado. Esta reducción de horas se ha producido debido a que el tiempo estimado se realizó pensando que había que realizar más vistas.
 - **Maquetación:** Esta desviación se ha producido por el mismo motivo que en el caso anterior. Al haber sólo una vista, el trabajo que hay que realizar en la maquetación es mucho menor que el estimado. Además, se han tenido que realizar muchas menos modificaciones de estilo que las estimadas, y de ahí que haya bastantes horas menos.
 - **Añadir funcionalidad:** En este caso, se ha necesitado emplear un mayor número de horas a la realización de la tarea ya que la ventana del administrador se ha utilizado bastante menos de lo estimado y ha sido necesario añadir demasiada funcionalidad desde 'back', lo que resulta mucho más costoso que trabajar con el administrador.

- **Pruebas:** Para esta tarea ha sido necesario emplear un mayor número de horas de lo estimado ya que a la hora de realizar las pruebas, se han detectado una serie de errores que a priori no tendrían que haberse producido. El resultado se ha reportado tras las correspondientes modificaciones en la aplicación.

Capítulo 8. Conclusiones

En este apartado, comentaré las conclusiones a las que he podido llegar una vez finalizado el proyecto. Además, plantearé algunas posibles mejoras.

8.1. Conclusiones

En primer lugar, tengo que decir que me llevo una muy buena experiencia en la empresa. Siempre que me ha surgido alguna duda, mis compañeros no han dudado un segundo en tratar de ayudarme. Además, la relación personal también ha sido muy satisfactoria.

Otro aspecto positivo que puedo destacar de esta experiencia es el aprendizaje del gestor de contenidos Drupal, el cual considero un buen CMS aunque contiene bastante funcionalidad y puede resultar muy difícil de llegar a comprender. También considero positivo el aprendizaje del lenguaje de programación PHP ya que es un lenguaje que se utiliza mucho a nivel global y nunca viene mal conocer lenguajes de programación nuevos. Además, he podido reforzar mis conocimientos de desarrollo web, sobre todo en javascript y peticiones AJAX, los cuales me han ayudado a aprobar una asignatura que me estaba dando numerosos problemas.

Un aspecto negativo que me llevo es cierta falta de tiempo para realizar el proyecto, ya que antes de empezar con el TFG tuve que terminar un módulo que permitiese la gestión de partes de trabajo, y sin el cual no podía realizar el proyecto. Este problema también se ha visto reflejado en lo apurado que he estado para poder terminar la memoria.

A pesar de este aspecto negativo, considero que ha sido una experiencia muy positiva, la cual me sirve para poder hacerme una idea del esfuerzo que conlleva realizar una aplicación de este tamaño y en un entorno de trabajo real.

8.2. Posibles mejoras

Al finalizar el proyecto, se me han ocurrido posibles mejoras de la aplicación, y que por falta de tiempo no he podido llegar a añadir.

Estas mejoras son las siguientes:

- Incorporación de nuevas gráficas al módulo, utilizando para ello un módulo de Drupal que ofrezca una mayor variedad de gráficas.
- Ofrecer al usuario la posibilidad de comparar los presupuestos de dos proyectos diferentes. Esto podría realizarse creando dos bloques de filtros, de tal manera que el primer bloque muestre una gráfica cuando se apliquen los filtros, y el segundo bloque muestre otra gráfica teniendo en cuenta los criterios de filtrado de este último.
- Otra posible mejora podría ser la opción de descarga de las gráficas pintadas en

formato .pdf y .xls.

- Por último, se podría modificar el módulo que controla los presupuestos de licitación y adjudicación, para que permita crear varios presupuestos de licitación y de adjudicación durante el transcurso del tiempo. De esta manera, se podrían pintar estos presupuesto agrupados por meses o por fechas. No obstante, considero que esta modificación sería bastante costosa en tiempo de desarrollo.

Capítulo 9. Bibliografía

La bibliografía empleada para realizar el desarrollo del proyecto ha sido la siguiente:

- Apuntes y trabajos realizados durante el grado.
- Manual de PHP
 - <http://php.net/manual/es/index.php>
- Manual de Drupal
 - <https://tutorial-drupal.com/>
- Manual de JQuery
 - <https://www.desarrolloweb.com/manuales/manual-jquery.html>
- Manual de AJAX
 - <http://www.uco.es/~lr1maalm/manualdeajax.pdf>
- Manual de uso Google Charts Tools
 - <https://www.youtube.com/watch?v=N9G2NpOjMbQ>